



**UNIVERSIDAD CARLOS III DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**

**INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN**

**PROYECTO DE FIN DE CARRERA**

# **GESTIÓN DE LA CALIDAD EN LOS REQUISITOS**

**Alumno: David Jiménez Mediero**

**Tutora UC3M: Anabel Fraga**

**Tutores de Empresa: Emiliano Fernández**

**José Miguel Fuentes**

**MES ,2015**

Esta página ha sido dejada en blanco intencionadamente

Título: Calidad en gestión de requisitos.

Autor: David Jiménez Mediero

Director:

## EL TRIBUNAL

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día \_\_ de \_\_\_\_\_ de 20\_\_  
en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda  
otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Esta página ha sido dejada en blanco intencionadamente

## Agradecimientos

Quiero agradecer a...

...mi tutora *Anabel Fraga*. Por entender cuál era mi situación y haber tenido tanta paciencia conmigo. También por tu ayuda y dedicación para sacar este proyecto adelante.

...mi tutor *José Miguel Fuentes*. Por tu ayuda desinteresada y por saber sacar tiempo para ayudarme aun cuando no disponías de él.

...mi tutor *Emiliano Fernández*. Por proponerme este proyecto, el cual se ha adaptado a mis inquietudes y mi disponibilidad.

...mis compañeras *Adriana García* y *Esther Labrador*. Porque si he sacado algo valioso de toda esta etapa aparte de todo lo aprendido, es vuestra amistad y todos los momentos que hemos compartido.

...mis padres *Pedro* y *Mª Carmen*. Por vuestro apoyo, vuestro ánimo y vuestra insistencia para que sacara esto adelante y sobre todo por haberme proporcionado los medios y la educación que me han convertido en la persona que soy hoy en día.

...mi pareja *Eva*. Por estar a mi lado y aguantarme cuando perdía los nervios o animarme cuando perdía los ánimos. Por celebrar mis éxitos y endulzarme los fracasos. Por todos estos años y experiencias juntos y por los que quedan por venir.

*“...algo termina, algo comienza.”*

*[Andrezj Sapkowski, Camino sin retorno]*



Esta página ha sido dejada en blanco intencionadamente

## **RESUMEN:**

Las especificaciones de requisitos son la base fundamental de todo desarrollo de un proyecto.

Una especificación de requisitos clara, concisa y correcta así como la correcta gestión de los requisitos evitan en gran medida la aparición de errores durante el transcurso de las fases del ciclo de vida del proyecto. Por lo tanto invertir tiempo y recursos en la gestión de la calidad de los requisitos evitará demoras en el tiempo y sobrecostos producidos por una mala realización y gestión de los requisitos.

Este proyecto fin de carrera ofrece una revisión sobre las metodologías, herramientas y compendios de buenas prácticas de las que disponen los desarrolladores para asegurar y gestionar la calidad de sus requisitos.

## ABSTRACT:

Requirements specifications are the main base of all project development.

A clear, concise and correct requirements specification and the proper requirements management largely avoid the occurrence of errors during the course of the phases of the project life cycle. Therefore, invest time and resources in requirements quality management will avoid time delays and budget overruns caused by poor implementation and management of requirements.

This project provides a review of methodologies, tools and compendia of good practices available to developers to ensure and manage the quality of their requirements specifications.



## Índice de contenidos:

Agradecimientos .....	5
RESUMEN: .....	7
ABSTRACT:.....	8
Índice de contenidos: .....	9
Índice de figuras: .....	12
1. Capítulo 1: Introducción:.....	15
1.1. Introducción a la gestión de requisitos: .....	15
1.2. Objetivos del proyecto fin de carrera: .....	16
1.3. Motivación: .....	16
1.4. Estructura del proyecto:.....	17
1.5. Planificación y presupuesto: .....	17
1.5.1. Introducción: .....	17
1.5.2. Planificación: .....	18
1.5.3. Diagrama Gantt: .....	19
1.6. Presupuesto: .....	20
2. Capítulo 2: Calidad y Gestión de Requisitos:.....	21
2.1. Ingeniería de Requisitos y Gestión de Requisitos: .....	21
2.2. Calidad en Gestión de Requisitos:.....	23
2.3. Buscando la calidad en la gestión de requisitos.....	26
2.3.1. Buenas Prácticas:.....	26
2.3.1.1. CMMI(“CMMI® para Desarrollo, Versión 1.3”): .....	26
2.3.1.1.1. CMMI - DESARROLLO DE REQUISITOS (Requirement Development RD).....	28
2.3.1.1.2. GESTIÓN DE REQUISITOS (Requirement Management REQM) .....	38
2.3.1.1.3. Conclusiones: .....	42
2.3.2. ITIL: .....	43
2.3.3. El proceso de revisión de requisitos:.....	48
2.3.4. Marcos y Normativas: .....	54
2.3.4.1. IEEE 830:1998: Recommended Practice for Software Requirements Specifications: .....	54

2.3.4.2.	ISO/IEC/IEEE 29148:2011: Systems and software engineering – Life cycle processes – Requirements engineering: .....	58
2.3.4.3.	ISO/IEC TR 24766 – Information technology – Systems and software engineering – Guide for requirements engineering tool capabilities: .....	63
2.3.4.4.	ISO/IEC 15288 – Systems and software engineering – Systems life cycle processes: .....	67
2.3.4.5.	ISO/IEC 12207 – Systems and software engineering – Software life cycle processes: .....	71
2.3.5.	Herramientas de creación y gestión de requisitos:.....	75
2.3.5.1.	Innoslate - Innovaciones SPEC ®:.....	76
2.3.5.1.1.	Funcionalidades principales: .....	76
2.3.5.1.1.1.	Creación de elementos de información en la BBDD: .....	76
2.3.5.1.1.2.	Panel de Control de Innoslate “Dashboard”: .....	86
2.3.5.1.1.3.	Menú de Requisitos:.....	87
2.3.5.1.1.4.	Generación de Diagramas: .....	89
2.3.5.1.2.	Gestión de la calidad en Innoslate: .....	89
2.3.5.1.3.	Conclusiones sobre Innoslate:.....	91
2.3.5.2.	Jama Product Delivery Platform - Jama Software: .....	92
2.3.5.2.1.	Funcionalidades principales: .....	92
2.3.5.2.1.1.	Pantalla inicial (Dashboard) :.....	92
2.3.5.2.1.2.	STREAM (Comentarios) : .....	93
2.3.5.2.1.3.	Projects (Gestión de proyectos) : .....	93
2.3.5.2.1.4.	Reviews (Revisiones) : .....	95
2.3.5.2.2.	Gestión de la calidad en Jama: .....	97
2.3.5.2.3.	Conclusiones sobre Jama: .....	97
2.3.5.3.	Requirements Quality Analyzer - The Reuse Company:.....	98
2.3.5.3.1.	Funcionalidades principales: .....	99
2.3.5.3.1.1.	Aseguramiento de la calidad:.....	99
2.3.5.3.1.2.	Gestión del conocimiento: .....	100
2.3.5.3.1.2.1.	Gestión de magnitudes por defecto:.....	100
2.3.5.3.1.2.2.	Gestión de sentencias especiales:.....	102
2.3.5.3.1.2.3.	Gestión de sustantivos de la ontología: .....	102
2.3.5.3.1.2.4.	Gestión de verbos de la ontología:.....	103
2.3.5.3.1.3.	Gestión del conocimiento: .....	104

2.3.5.3.1.3.1.	Asignación de métricas a hojas: .....	104
2.3.5.3.1.3.2.	Métricas de hojas: .....	105
2.3.5.3.1.3.3.	Magnitudes de libro: .....	106
2.3.5.3.1.4.	Control de calidad: .....	107
2.3.5.3.1.4.1.	Control de calidad – Cuadro de mando:.....	107
2.3.5.3.1.4.2.	Control de calidad - Métricas: .....	110
2.3.5.3.1.4.3.	Control de calidad - Usuarios: .....	111
2.3.5.3.1.4.4.	Control de calidad – Conflictos de Unidades: .....	112
2.3.5.3.1.4.5.	Control de calidad – Solapamiento: .....	113
2.3.5.3.1.4.6.	Control de calidad – Completitud: .....	113
2.3.5.3.2.	Conclusiones sobre RQA: .....	114
3.	Capítulo 3: Parte práctica – Cuestionario sobre calidad de requisitos: .....	116
3.1.	Introducción: .....	116
3.2.	Estructura del cuestionario: .....	116
3.2.1.	Características de la empresa: .....	116
3.2.2.	Proceso de captura de requisitos:.....	117
3.2.3.	Verificación y validación de requisitos:.....	118
3.2.4.	Gestión de los requisitos:.....	118
3.2.5.	Herramientas para requisitos:.....	119
3.3.	Exposición de los resultados del cuestionario: .....	120
3.3.1.	Datos sobre características de la empresa:.....	120
3.3.2.	Datos sobre el proceso de captura de requisitos:.....	121
3.3.3.	Datos sobre procesos de verificación y validación de los requisitos: .....	122
3.3.4.	Datos sobre procesos de gestión de los requisitos: .....	124
3.3.5.	Datos el uso de herramientas para la gestión de los requisitos: .....	125
3.4.	Conclusiones sobre el cuestionario:.....	127
4.	Conclusiones y trabajo a futuro: .....	128
4.1.	Conclusiones generales:.....	128
4.2.	Trabajo a futuro: .....	129
5.	Glosario de términos:.....	130
6.	Bibliografías y referencias: .....	132

## Índice de figuras:

Figura 1: Planificación de tareas .....	18
Figura 2: Diagrama Gantt del proyecto .....	19
Figura 3: Presupuesto estimado de realización del proyecto .....	20
Figura 4: Actividades en la Ingeniería de Requisitos [2] .....	21
Figura 5: Proporción de Errores en cada una de la fases de desarrollo [2] .....	25
Figura 6: Evolución temporal del coste de Mantenimiento [2] .....	25
Figura 7: Resultados de los Informe CHAOS [3] .....	26
Figura 8: Resumen tratamiento Requisitos en CMMI v1.3 .....	43
Figura 9: Relaciones entre procesos y requisitos [2].....	46
Figura 10: Relaciones entre productos y procesos en ITIL v3 [2].....	47
Figura 11: Fases del proceso de inspección (Revisión de Requisitos).....	51
Figura 12: Ejemplo de lista de control de defectos [8]. .....	53
Figura 13: Ejemplo de documento StRS [13].....	61
Figura 14: Ejemplo de documento SyRS [13]. .....	62
Figura 15: Ejemplo de documento SRS [13].....	63
Figura 16: Pantalla de creación de entidades de Innoslate. ....	77
Figura 17: Pantalla de creación de “Actions” de Innoslate. ....	77
Figura 18: Pantalla de creación de “Artifacts” de Innoslate. ....	78
Figura 19: Pantalla de creación de “Assets” de Innoslate.....	78
Figura 20: Pantalla de creación de “Characteristics” de Innoslate. ....	79
Figura 21: Pantalla de creación de “Conections” de tipo “Conduit” de Innoslate.....	79
Figura 22: Pantalla de creación de “Conections” de tipo “Logical” de Innoslate. ....	80
Figura 23: Pantalla de creación de “Costs” de Innoslate. ....	80
Figura 24: Pantalla de creación de “Decisions” de Innoslate.....	81
Figura 25: Pantalla de creación de “Inputs/Outputs” de Innoslate. ....	81
Figura 26: Pantalla de creación de “Locations” de tipo “Orbital” de Innoslate.....	82
Figura 27: Pantalla de creación de “Locations” de tipo “Physical” de Innoslate. ....	82
Figura 28: Pantalla de creación de “Locations” de tipo “Virtual” de Innoslate. ....	83
Figura 29: Pantalla de creación de “Measures” de Innoslate. ....	83
Figura 30: Pantalla de creación de requisitos de Innoslate. ....	84
Figura 31: Pantalla de creación de “Resources” de Innoslate. ....	85
Figura 32: Pantalla de creación de “Risks” de Innoslate. ....	85
Figura 33: Pantalla de creación de “Statements” de Innoslate. ....	86
Figura 34: Pantalla de creación de “Time” de Innoslate. ....	86
Figura 35: Panel de Control “Dashboard” de Innoslate. ....	87
Figura 36: Selección de documentos de requisitos.....	88
Figura 37: Vista general de los requisitos. ....	88
Figura 38: Ejemplo diagrama de tipo árbol.....	89
Figura 39: Sugerencias tras análisis de un requisito. ....	90
Figura 40: Página de Inicio de Jama. ....	93
Figura 41: Comentario en ventana de STREAM. ....	93

Figura 42: Creación de elementos en Jama .....	94
Figura 43: Cuadros de utilidades de los elementos .....	95
Figura 44: Selección de requisitos para revisión.....	96
Figura 45: Opciones para los aprobadores de requisitos.....	96
Figura 46: Configuración de métricas .....	99
Figura 47: Configuración de reglas en las métricas.....	100
Figura 48: Gestión de magnitudes .....	101
Figura 49: Configuración de sentencias especiales.....	102
Figura 50: Sustantivos de la ontología .....	103
Figura 51: Verbos de la ontología .....	104
Figura 52: Asignación de métricas a hojas .....	105
Figura 53: Configuración de métricas en cada hoja .....	105
Figura 54: Grupos de patrones.....	106
Figura 55: Configuración de magnitudes del libro .....	107
Figura 56: Control Calidad - Cuadro de mando.....	108
Figura 57: Vista de Calidad: Conceptos fuera de dominio .....	109
Figura 58: Vista de Calidad: Uso de voz pasiva .....	109
Figura 59: Vista de Calidad - Detalle recomendación métricas .....	110
Figura 60: Control de calidad: Métricas .....	111
Figura 61: Control de calidad: Usuarios .....	112
Figura 62: Control de calidad - Conflictos de Unidades .....	112
Figura 63: Control de calidad - Solapamientos .....	113
Figura 64: Control de calidad - Completitud .....	114
Figura 65: Cuestionario - Características de la empresa .....	117
Figura 66: Cuestionario - Captura de requisitos.....	117
Figura 67: Cuestionario - Verificación y validación .....	118
Figura 68: Cuestionario - Gestión de requisitos .....	118
Figura 69: Cuestionario - Herramientas para requisitos .....	119
Figura 70: Empleados y edad de las empresas .....	120
Figura 71: Sectores de mercado en los que participan .....	120
Figura 72: Métodos utilizados para la elaboración de requisitos .....	122
Figura 73: Reutilización de requisitos .....	122
Figura 74: Aplicación de fase de revisión .....	123
Figura 75: Distribución de los participantes en los procesos de revisión .....	123
Figura 76: Modo realización proceso de verificación y validación.....	124
Figura 77: Tipos de herramientas y utilización .....	125
Figura 78: Herramientas más utilizadas .....	126



# 1. Capítulo 1: Introducción:

## 1.1. Introducción a la gestión de requisitos:

Un buen producto, bajo el criterio del usuario, es aquél que ha sido creado por desarrolladores que han tenido muy claro qué era lo que se pretendía conseguir y cómo se tenía que actuar para obtenerlo. Para desarrollar un buen producto es fundamental comprender a fondo las necesidades del usuario, se debe entender su trabajo, cómo facilitará el producto el desarrollo de sus actividades y cómo se adecuará a los objetivos y posibilidades de la organización tanto desarrolladora como destinataria del producto. Para llevar a cabo estas tareas con éxito es fundamental realizar una captura y gestión de requisitos en las que la búsqueda de la calidad y el refinamiento del proceso sea una máxima durante todo el ciclo de vida del producto.

El coste asociado a una recogida de requisitos de calidad y al análisis del sistema a desarrollar es menor comparado con el coste resultante de tener requisitos pobres; es decir, el coste de reparar productos deficientes o de poca calidad, el coste de los proyectos cancelados y el coste de haber perdido la oportunidad de tener el producto correcto en el momento correcto siempre será mucho mayor que el coste de tiempo y recursos invertidos en la correcta recogida y gestión de los requisitos del sistema que se va a desarrollar. Realizar una recogida de requisitos exhaustiva y bien documentada asegurará conocer el alcance real de lo que se debe realizar y minimizar el número de riesgos en un futuro, por lo tanto es esfuerzo bien invertido tanto de cara al cliente como de forma interna al tener un proceso controlado y monitorizado en cada una de sus fases.

Tal y como describe el Instituto Nacional de Tecnologías de Comunicación (INTECO) en [1], el fundamento básico de cualquier software recae sobre su proceso de ingeniería de requisitos. El éxito o fracaso del software depende casi siempre de cómo de bien se hayan capturado, entendido y usado los requisitos como base para el desarrollo. La ingeniería de requisitos es parte de la ingeniería del software donde se definen las propiedades y la estructura del mismo. La ingeniería de requisitos comprende el desarrollo y gestión de requisitos.

- El **desarrollo de requisitos** implica entender los requisitos de negocio, identificar los requisitos de usuario y trasladar los requisitos de usuario y de negocio a requisitos de sistema/software.

-La **gestión de requisitos** implica gestionar los cambios de requisitos y mantener la consistencia entre los requisitos y otros productos de trabajo del proyecto.

La Gestión de Requisitos en Ingeniería de Sistemas, es el proceso encargado de la identificación, asignación y seguimiento de los requisitos, incluyendo el interfaz, verificación, modificación y control del estatus a lo largo del ciclo de vida. Es el conjunto de actividades que se concentra en el aseguramiento de las especificaciones, por ejemplo, los requisitos que son reunidos para la satisfacción del cliente. Es el proceso que inicia con la concepción de un proyecto y continúa hasta el resultado final del producto(s), donde después de esto ya no es necesario.

Los cambios de requisitos deben ser gestionados para asegurar que la calidad de los mismos se mantengan, los problemas suscitados por los cambios de requisitos podrían incurrir en altos costos, siendo los requisitos un factor crítico de riesgos.

## **1.2. Objetivos del proyecto fin de carrera:**

El objetivo de este trabajo es realizar un estudio sobre los marcos, técnicas y buenas prácticas cuya aplicación asegura aumentar el nivel de la calidad en la gestión de requisitos de un proyecto software. También se realizará una revisión sobre algunas de las herramientas para el tratamiento y gestión de requisitos disponibles en el mercado que ayudan a alcanzar buenos niveles de calidad con su utilización. Se expondrá qué se entiende como gestión de requisitos, algunas de las metodologías de buenas prácticas existentes para la gestión su gestión y correcta realización, algunos de los marcos y normativas acerca de requisitos y las revisiones sobre 3 herramientas para el tratamiento de requisitos cada una con un enfoque diferenciado.

Como parte práctica del proyecto se realizará un auto-cuestionario o *self-assessment* dirigido a empresas cuya finalidad es determinar hasta qué punto las compañías cumplen con las buenas prácticas indicadas en los métodos y normativas incluidos en este trabajo. Una vez obtenidas las respuestas a dicha encuesta se expondrán los resultados en este documento.

Esta encuesta será elaborada en castellano e inglés y será dirigida hacia empresas nacionales e internacionales pertenecientes a diferentes sectores con el objetivo de lograr un mayor espectro de encuestados y visualizar los diferentes enfoques de cada empresa dependiendo del tipo de negocio al que se dedican.

## **1.3. Motivación:**

Tras haber participado en varios proyectos para diferentes empresas y clientes y experimentar los diferentes problemas en cada una de sus fases, me queda claro que un punto clave para terminar un proyecto con éxito es una correcta captura y gestión de los requisitos. Los requisitos conforman el punto de nacimiento de cualquier proyecto el cual parte de unas necesidades determinadas. A menudo las organizaciones y empresas descuidan este aspecto no dedicándole el tiempo y esfuerzo necesarios por ser una etapa del proyecto que no produce ningún aporte o avance visible para los clientes. Dentro de un mercado en el que los tiempos y presupuestos están ajustados al máximo, dedicar una gran parte del tiempo a esta etapa puede ser difícilmente justificable para la mayor parte de los clientes. Por esta misma razón es importante resaltar que cuidar e invertir tiempo en los requisitos no sólo aporta una correcta captura y gestión de las necesidades, si no que ayuda a tener una visión exacta del alcance y el estado del proyecto en cualquier momento de su desarrollo y evita en gran medida demoras en los tiempos y presupuestos preestablecidos.



## **1.4. Estructura del proyecto:**

Este Proyecto Final de Carrera, está dividido en dos partes, una parte teórica y una práctica.

En la parte teórica “Capítulo 2: Calidad y Gestión de Requisitos” se hace una exposición de los medios disponibles para asegurar que los requisitos son realizados y gestionados con una calidad suficiente a lo largo del ciclo de vida del proyecto. Dentro de este apartado se expondrá qué es un requisito y como afecta el nivel de su calidad en el proyecto, una serie de buenas prácticas y metodologías destacando *CMMI* e *ITIL*, una selección de normativas sobre la calidad en los requisitos y una revisión de 3 herramientas disponibles en el mercado para analizar y gestionar la calidad en los requisitos.

En la parte práctica se realizará una encuesta de evaluación dirigida a varias empresas y organizaciones para estudiar su implicación y los métodos que utilizan respecto a la gestión de la calidad de sus requisitos y se expondrán los resultados obtenidos.

## **1.5. Planificación y presupuesto:**

### **1.5.1. Introducción:**

En el siguiente apartado se muestra la planificación del proyecto y el presupuesto del mismo.

El cálculo de la planificación y del presupuesto se han formulado en base a las siguientes premisas:

- Meses laborables al año = 12 meses.
- Días laborables al mes = 20 días.
- Horas diarias de trabajo = 2 horas.
- La realización del proyecto se ha estimado en un total de 788 horas.

Tanto la planificación como el diagrama de Gantt han sido realizadas con la herramienta MS Project 2010.

## 1.5.2. Planificación:

En la figura 1 se representa la planificación del proyecto mostrando su duración, fecha de inicio, fecha de finalización y sus dependencias.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	<b>PFC - Calidad en Gestión de Requisitos</b>	394 días	lun 24/11/14	mié 26/08/15	
2	<b>Normativas para requisitos y metodologías de buenas prácticas</b>	110 días	lun 24/11/14	lun 09/02/15	
3	Conseguir y reunir documentación sobre normativas de requisitos y metodologías de buenas prácticas	50 días	lun 24/11/14	lun 29/12/14	
4	Lectura y estudio de la documentación	60 días	mar 30/12/14	lun 09/02/15	3
5	<b>Herramientas para la gestión de la calidad en los requisitos</b>	70 días	mar 10/02/15	lun 30/03/15	
6	Estudio y elaboración de proyectos de ejemplo sobre las herramientas de gestión de la calidad de requisitos	70 días	mar 10/02/15	lun 30/03/15	4
7	<b>Cuestionario sobre uso de medidas para la gestión de la calidad de los requisitos en empresas</b>	50 días	jue 02/07/15	jue 06/08/15	
8	Realización del cuestionario	30 días	jue 02/07/15	jue 23/07/15	20
9	Obtención de resultados	10 días	jue 30/07/15	jue 06/08/15	27
10	<b>Memoria del PFC</b>	214 días	mar 31/03/15	mié 26/08/15	
11	Creación del documento, estructura y resumen	10 días	mar 31/03/15	lun 06/04/15	6
12	<b>Capítulo I: Introducción</b>	190 días	mar 07/04/15	lun 17/08/15	
13	Introducción, objetivos y motivación	15 días	mar 07/04/15	jue 16/04/15	11
14	Estructura del proyecto, Gantt y presupuesto	7 días	mié 12/08/15	lun 17/08/15	25
15	<b>Capítulo II: Calidad y gestión de requisitos</b>	110 días	jue 16/04/15	jue 02/07/15	
16	Introducción del capítulo II	15 días	jue 16/04/15	lun 27/04/15	13
17	Buenas prácticas	20 días	mar 28/04/15	lun 11/05/15	16
18	Revisión de requisitos	10 días	mar 12/05/15	lun 18/05/15	17
19	Normativas	30 días	mar 19/05/15	lun 08/06/15	18
20	Herramientas	35 días	mar 09/06/15	jue 02/07/15	19
21	<b>Capítulo III: Caso práctico - Cuestionario sobre calidad de requisitos</b>	26 días	jue 23/07/15	mar 11/08/15	
22	Introducción y estructura del cuestionario	8 días	jue 23/07/15	mié 29/07/15	8
23	Exposición de resultados	6 días	jue 06/08/15	mar 11/08/15	9
24	<b>Conclusiones</b>	2 días	mar 11/08/15	mié 12/08/15	
25	Conclusiones del proyecto	2 días	mar 11/08/15	mié 12/08/15	23
26	<b>Bibliografía</b>	2 días	mié 29/07/15	jue 30/07/15	
27	Elaboración de la bibliografía	2 días	mié 29/07/15	jue 30/07/15	22
28	Revisión y correcciones 1	5 días	mar 18/08/15	jue 20/08/15	14
29	Revisión y correcciones 2	5 días	jue 20/08/15	lun 24/08/15	28
30	Revisión de formato y agradecimientos	4 días	mar 25/08/15	mié 26/08/15	29
31	<b>Presentación del PFC</b>	10 días	jue 27/08/15	mié 02/09/15	
32	Realización de diapositivas	10 días	jue 27/08/15	mié 02/09/15	30
33	Entrega final	1 día	jue 03/09/15	jue 03/09/15	32

Figura 1: Planificación de tareas



1.5.3. Diagrama Gantt:

En la figura 2 se muestra el diagrama Gantt de la planificación de las tareas del proyecto.

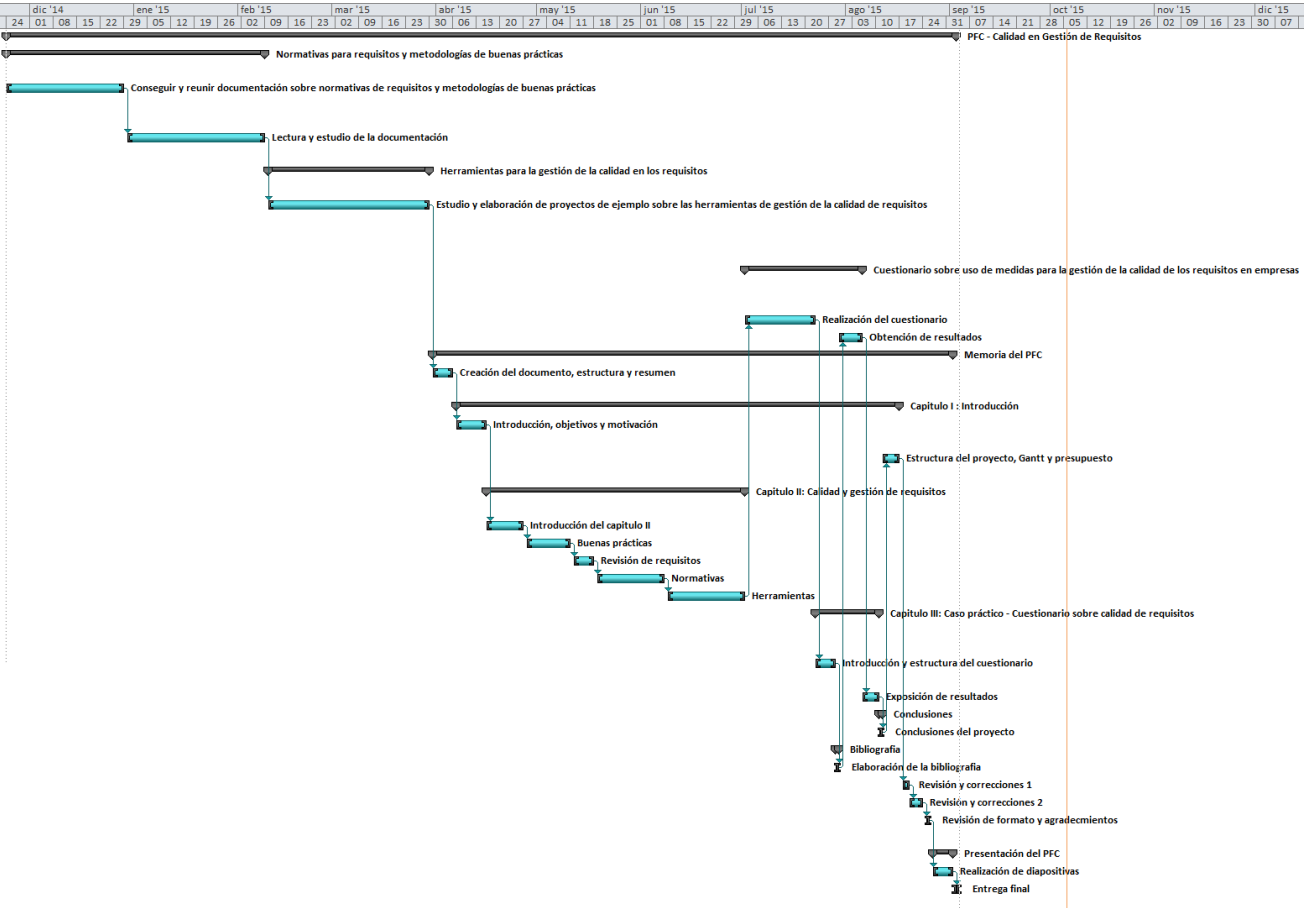


Figura 2: Diagrama Gantt del proyecto

## 1.6. Presupuesto:

A continuación, en la figura 3, se ofrece el presupuesto estimado para el desarrollo de este proyecto.



UNIVERSIDAD CARLOS III DE MADRID  
Escuela Politécnica Superior

### PRESUPUESTO DE PROYECTO

**1.- Autor:**  
David Jiménez Mediero

**2.- Departamento:**  
Departamento de Informática

**3.- Descripción del Proyecto:**  
- Título: Gestión de la calidad en los requisitos  
- Duración (meses): 10  
- Tasa de costes Indirectos: 20%

**4.- Presupuesto total del Proyecto (valores en Euros):**  
Euros

**5.- Desglose presupuestario (costes directos)**

**PERSONAL**

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación <sup>a)</sup> (hombres mes)	Coste hombre mes	Coste (Euro)
David Jiménez		Ingeniero	4,95	2.694,39	13.337,23
Anabel Fraga		Ingeniero Senior	0,57	4.289,54	2.445,04
José Miguel Fuentes		Ingeniero Senior	0,38	4.289,54	1.630,03
Emiliano Fernández		Ingeniero Senior	0,19	4.289,54	815,01
					0,00
Hombres mes 6,09				Total	18.227,31

<sup>a)</sup> 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)  
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

**EQUIPOS**

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable <sup>d)</sup>
PcCom Evolution FX-9370	1.180,00	100	10	60	196,67
Lenovo IdeaPad Z50-70	690,00	100	10	60	115,00
		100		60	0,00
		100		60	0,00
		100		60	0,00
					0,00
Total					311,67

<sup>d)</sup> Fórmula de cálculo de la Amortización:  

$$\frac{A}{B} \times C \times D$$
A = nº de meses desde la fecha de facturación en que el equipo es utilizado  
B = periodo de depreciación (60 meses)  
C = coste del equipo (sin IVA)  
D = % del uso que se dedica al proyecto (habitualmente 100%)

**SUBCONTRATACIÓN DE TAREAS**

Descripción	Empresa	Coste imputable
Total		0,00

**OTROS COSTES DIRECTOS DEL PROYECTO<sup>e)</sup>**

Descripción	Empresa	Costes imputable
Compra de documentación de normativas	<a href="http://www.iso.org">http://www.iso.org</a>	198,50
Material de oficina		100,00
Total		298,50

<sup>e)</sup> Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

**6.- Resumen de costes**

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	18.227
Amortización	312
Subcontratación de tareas	0
Costes de funcionamiento	299
Costes Indirectos	3.767
Total	22.605

Figura 3: Presupuesto estimado de realización del proyecto

## 2. Capítulo 2: Calidad y Gestión de Requisitos:

### 2.1. Ingeniería de Requisitos y Gestión de Requisitos:

La necesidad de la ingeniería de requisitos dentro de la ingeniería del software es imprescindible para obtener productos de calidad. La calidad no puede aplicarse en las etapas finales de los procesos de desarrollo sino que es una característica del propio producto, iniciándose y aplicándose desde la primera etapa en las especificaciones de los productos.

Posibles definiciones del concepto de requisito las encontramos en diversas publicaciones, donde se define el término “requisito” como una especificación de lo que debe ser implementado. Son descripciones de cómo el sistema se debe comportar, o las propiedades o atributos que el sistema debe contener. Los requisitos deben ser una restricción en el proceso de desarrollo del sistema.

Dentro de los movimientos más relevantes que en la ingeniería del software han ido surgiendo en los últimos tiempos podemos citar a la necesidad del modelado de los procesos de negocio y la interoperabilidad entre sistemas. La ingeniería de requisitos debe responder a estas necesidades como parte del ciclo de desarrollo de los sistemas.

Con este enfoque, el giro a la definición de los requisitos como los servicios que el sistema debe proporcionar junto a las restricciones en la operación del sistema; se establece más acorde a estos nuevos enfoques.

En [2] se describen las actividades de las que consta la Ingeniería de requisitos tal y como se muestran en la figura 4 y se describen más adelante.

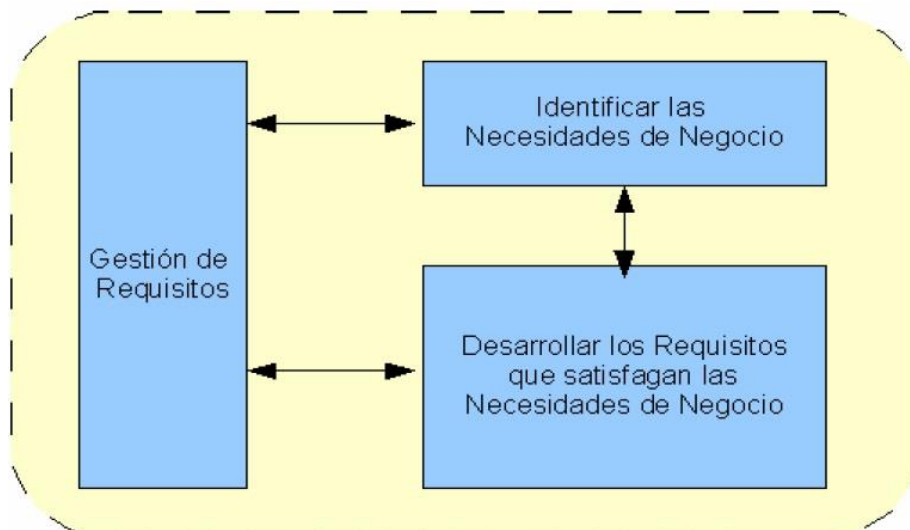


Figura 4: Actividades en la Ingeniería de Requisitos [2]

Estas actividades se desarrollan a lo largo de todo el ciclo de vida del producto, ya que las necesidades del negocio van cambiando conforme el producto se explota en producción. La

actividad de gestión de requisitos conlleva una serie de actividades dentro de la gestión de proyectos a nivel global, como son la gestión del cambio, la creación de líneas base, incluyendo aspectos financieros del propio proyecto. Es decir, una ampliación de funcionalidad o error encontrado en el producto debe ser analizado en relación a cuáles han sido las causas, el impacto de la eliminación del error o ampliación de funcionalidad que tiene en el proyecto, tiempo de resolución, coste y presupuesto disponibles, comprobar que el servicio o parte del proceso no es ofrecido por la organización, comprobar que el cambio está en concordancia con las líneas estratégicas y las necesidades del negocio de la organización, etc. La ingeniería de requisitos no debe ser considerada meramente como una elicitación de requisitos aislada en las fases iniciales del proyecto, sino que exige una retroalimentación continua de la organización. Por todos estos motivos, la ingeniería de requisitos debe reforzar los dos aspectos anteriormente citados: el modelado de procesos y la interoperabilidad.

Entendemos entonces como Gestión de Requisitos a aquellas tareas que registran y mantienen los requisitos en su constante evolución durante el ciclo de vida del proyecto y a las tareas que registran y mantienen la información histórica y de contexto de las actividades de ingeniería de requisitos.

Con este enfoque, dentro de las actividades generales en la gestión de requisitos, podemos desglosar las actividades principales en:

- **Identificar las necesidades del negocio:**

- **Estudiar el dominio del problema.** Su objetivo principal es obtener un conocimiento suficiente del dominio del problema como para poder comunicarse eficazmente con clientes y usuarios, comprender su negocio, entender sus necesidades y poder proponer una solución adecuada.

- **Identificar aspectos positivos y negativos de la situación actual.** Su objetivo principal es hacer explícitos qué aspectos funcionan bien y cuales no antes de la implantación del nuevo sistema. Los aspectos negativos identificados ofrecen la posibilidad de evitar que ocurran en el sistema a desarrollar y los positivos para conservarlos en la medida de lo posible.

- **Estudiar el modelo de negocio del cliente.** Su objetivo principal es comprender el negocio actual, especialmente sus procesos de negocio. Independientemente de que los procesos de negocio actuales sean manuales o estén automatizados, esta tarea es fundamental para entender el contexto en el que se usará el sistema a desarrollar y promover posibles mejoras.

- **Estudiar el entorno tecnológico del cliente.** Su objetivo es conocer el entorno tecnológico actual, tanto el *hardware* y el *software*, redes, etc., como la *arquitectura de servicios* que pueda estar implantada.

- **Obtener y documentar las necesidades de clientes y usuarios.** Su objetivo principal es obtener las *necesidades de clientes y usuarios* y documentarlas como *objetivos de negocio*, incluyendo los *modelos de procesos del negocio a implantar* si difieren significativamente del modelo de negocio actual, o si no se consideró oportuno realizar el estudio del negocio actual previamente en el proyecto.

- **Desarrollar los requisitos de un sistema software que satisfaga las necesidades del negocio:**

- **Desarrollar la visión general del sistema.** Su objetivo principal es comenzar a definir el sistema software (la *solución*) que satisfaga las necesidades de negocio de clientes y usuarios (el *problema*). Para ello, se definen los *requisitos generales* y se comienzan a especificar los *casos de uso del sistema* en su versión inicial. Conforme avance el proceso de ingeniería de requisitos, los requisitos generales se irán detallando en requisitos más específicos y los casos de uso que se considere oportuno irán evolucionando hacia su forma detallada.

- **Documentar los requisitos del sistema.** Su objetivo principal es continuar con la definición del sistema software a desarrollar, tomando como punto de partida los *requisitos generales* y los *casos de usos en su versión inicial*, y considerando los *objetivos de negocio* y el

*modelo de negocio a implantar*. Los casos de uso que se considere necesario, se van completando con más información y los requisitos generales se van detallando en *requisitos funcionales, no funcionales, de integración y en restricciones técnicas*.

- **Definir las posibilidades de integración del sistema.** Su objetivo principal es identificar sistemáticamente los servicios en funcionamiento o que estén en desarrollo en el entorno de la organización y que pueden tener impacto en el sistema a desarrollar.
- **Analizar los requisitos del sistema.** Su objetivo principal es analizar los requisitos previamente elicitados y documentados para detectar posibles conflictos.
- **Verificar la calidad de los requisitos del sistema.** Su principal objetivo es la comprobación de si los requisitos se están elaborando correctamente.
- **Validar los requisitos del sistema.** Su objetivo principal es la comprobación de la concordancia entre los requisitos elicitados corresponden a las necesidades de usuarios y clientes.

- **Gestionar los requisitos del sistema software a desarrollar:**

- **Definir el ciclo de vida de los requisitos del sistema.** Su objetivo principal es definirlos estados por los que los requisitos van a ir pasando. Es necesario estos estados para controlar las actividades pertenecientes al desarrollo de requisitos.
- **Gestionar las líneas base de los requisitos del sistema.** Su objetivo fundamental es la generación de versiones del proyecto. Es decir, controlar la consistencia del producto en un instante dado.
- **Gestionar los conflictos de los requisitos.** Su objetivo principal es la eliminación de posibles conflictos en el desarrollo de los requisitos.
- **Gestionar las peticiones de cambio.** Su objetivo principal es gestionar todas las peticiones de cambio en los requisitos una vez generada la primera *línea base* de la *Especificación de Requisitos del Sistema*, asegurándose que se sigue el procedimiento de control de cambios establecido en el proyecto y que por lo tanto, puede ser necesaria una modificación en los costes, recursos y fecha de entrega inicialmente acordados.
- **Gestionar la trazabilidad de los requisitos del sistema.** Su objetivo fundamental es comprobar la implementación de los requisitos y el grado de impacto de los cambios.

En la documentación de los requisitos, se deben tener las siguientes recomendaciones para una buena definición de requisitos en las organizaciones:

- Estructurar los requisitos. Problema de los requisitos duplicados u omitidos.
- Gestionar los requisitos: base para los contratos
- Gestionar las restricciones: rendimiento, seguridad, etc.
- Realizar pruebas de los requisitos
- Gestionar el control de cambios de los requisitos
- Obtener métricas de la evolución de los mismos
- Proporcionar ejemplos
- Reutilizar requisitos

## 2.2. Calidad en Gestión de Requisitos:

Resulta erróneo creer que realizar una especificación de requisitos muy completa es más que suficiente para cubrir esa etapa del ciclo de vida. Si una vez obtenidos se descuida el tratamiento de los requisitos o si no se pone el empeño y cuidado adecuados en su gestión podemos sufrir los mismos o peores problemas que si no hubiéramos realizado una especificación adecuada, con lo cual no solamente hay que invertir tiempo y esfuerzo en realizar una captura de requisitos exhaustiva sino que éstos tienen que tener un tratamiento posterior con un determinado nivel de calidad para resultar efectivos en el desarrollo de los productos software.

En [1] se incluyen estas 5 pautas generales sobre cómo realizar la gestión de requisitos de modo que se cumplan unos criterios mínimos de calidad:

#### **1º Entender los requisitos:**

Es importante entender cada requisito en toda su extensión, analizarlo, consultar la documentación asociada al requisito, revisar el requisito con quién lo ha identificado y de fijar bien las condiciones de cumplimiento o no de dicho requisito.

#### **2º Obtener el compromiso con los requisitos:**

Hay que ser conscientes en todo momento del estado de los requisitos, tanto en lo concerniente a los nuevos requisitos que van surgiendo a lo largo del proyecto como a los cambios que sufre cada uno. Es necesario documentar correctamente cada requisito anotando de dónde surge el requisito o la modificación sobre un requisito, quien lo aprueba, quien lo prueba, quien lo desarrolla y quien lo acepta. Realizando esto siempre tendremos una visión precisa del estado exacto del proyecto con cada cambio que vaya surgiendo. Un requisito no debe incorporarse hasta que no haya sido mínimamente negociado.

#### **3º Gestionar los cambios en los requisitos:**

Los cambios en los requisitos son necesarios, puesto que enriquecen y mejoran el proyecto a lo largo de su ciclo de vida, con lo cual hay que mantenerse abierto a ellos. No obstante es muy importante ser consciente del impacto que tiene cada modificación de los requisitos en el proyecto. Todos los requisitos han de ser versionados para poder consultar su estado en cada momento del proyecto.

#### **4º Mantener la trazabilidad bidireccional de los requisitos:**

Es conveniente saber en cada momento de dónde viene y adónde va cada requisito. Los requisitos técnicos deben tener su correspondencia con alguna línea o punto del documento de especificaciones que envía el cliente. Además, conviene que la relación entre los propios requisitos técnicos esté a su vez documentada y que se sepa de dónde vienen los productos de trabajo. Una forma habitual de afrontar todo esto es mediante el uso de diversas herramientas para gestión de requisitos. Si mantenemos bien definida y actualizada la trazabilidad, resultará más sencillo saber qué elementos se ven impactados cuando ocurre algún cambio. La trazabilidad se puede llevar a varios niveles, en función de tus necesidades.

**5º Asegurar que los productos de trabajo cumplan con los requisitos:** Realizar auditorías periódicas asegura que lo que se está desarrollando está de acuerdo con lo que se ha pedido. A partir de estas se deben lanzar acciones correctivas a las inconsistencias detectadas. Estas auditorías deben ser realizadas por alguien independiente al desarrollo de los requisitos a evaluar para garantizar una revisión objetiva.

Como se describe en [2] desde hace tiempo se conocen los datos del alto coste que implica reparar un error o defecto una vez realizada la entrega del producto. En comparación con lo que hubiera implicado detectarlo en fases más tempranas del desarrollo, varios estudios nos muestran una oscilación entre un 60% y un 100 % de sobre coste.

También se conoce desde hace tiempo que el 56% del total de errores encontrados en los productos software, son generados en la fase de Requisitos. El tanto por ciento del esfuerzo total del proyecto que se destina en la fase de Requisitos no llega al 20% independientemente del modelo de desarrollo (iterativo, en cascada, etc) o del tipo de proyecto (nuevo, de mantenimiento). En la figura 5 se muestran el porcentaje de errores encontrados en cada fase de desarrollo.



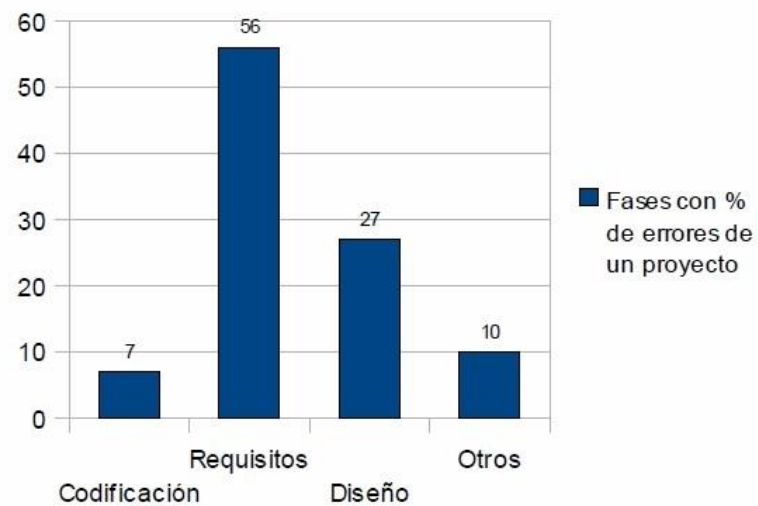


Figura 5: Proporción de Errores en cada una de la fases de desarrollo [2]

Si además se tiene cuenta los gastos invertidos en el mantenimiento del software, obtenidos en [2] y mostrados en la figura 6, es indudable que el problema persiste o es mejorable.

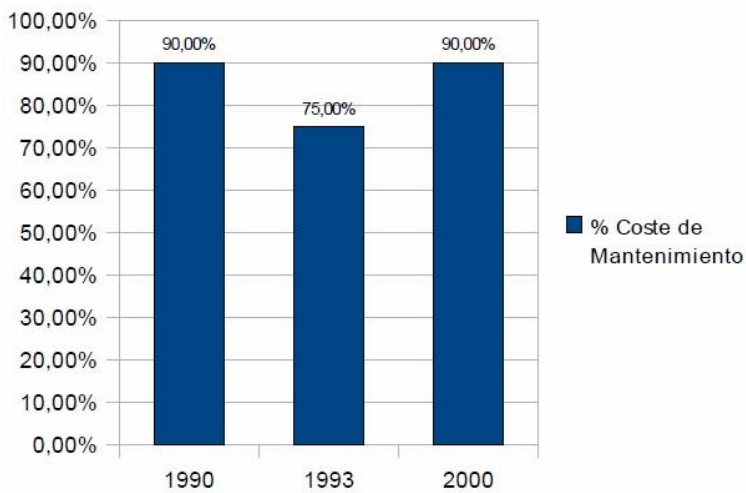


Figura 6: Evolución temporal del coste de Mantenimiento [2]

Las acciones llevadas a cabo para mejorar esta situación han sido diversas: desde nuevos lenguajes, metodologías en gestión de proyectos, nuevos paradigmas como la Orientación a Servicios, etc. Estas actuaciones avalan los avances producidos en la disciplina de Ingeniería del Software. Como puede apreciarse, en la figura 7 se observan las curvas de tendencias de los resultados de los proyectos de los datos ofrecidos por los informes CHAOS.

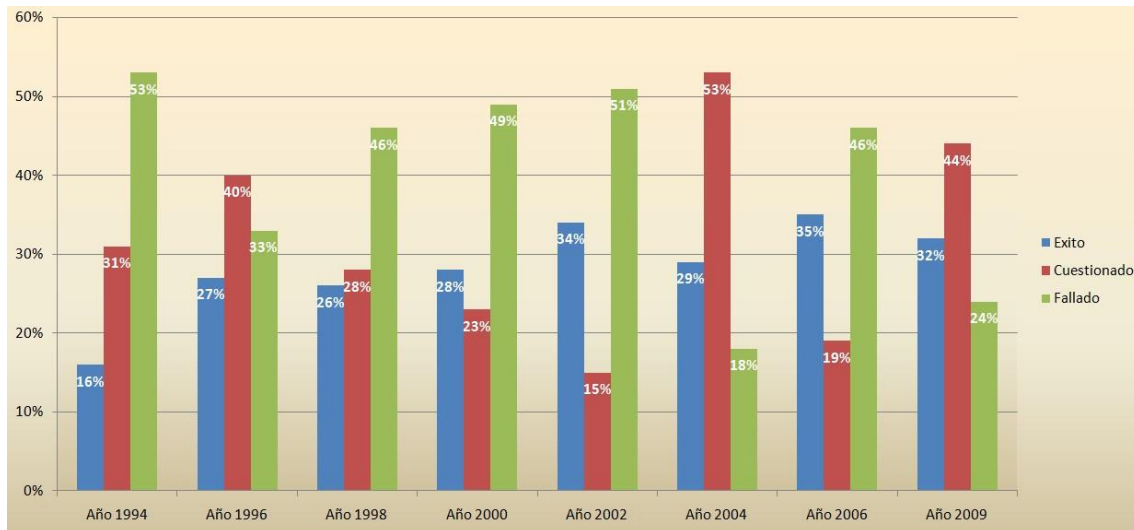


Figura 7: Resultados de los Informe CHAOS [3]

Se define como un proyecto realizado con éxito, aquellos que cumplen con el alcance definido, son terminados en tiempo y con el coste de los datos con los que fueron planificados. Se definen como proyectos cancelados, aquellos que no llegaron a terminarse. Proyectos cuestionados son aquellos que fueron modificados en alguno de estos parámetros: alcance inicial, tiempo estimado y/o coste planificado.

Como puede valorarse en los resultados de los Informes CHAOS obtenidos en [3] el nivel de fracaso y modificación de alcance de los proyectos es muy elevado con lo cual es vital una gestión de requisitos de calidad para conseguir el éxito. Por otra parte cabe destacar que en proyectos de áreas como el sector aeroespacial, naval, militar o sanidad no cuidar la calidad de un requisito puede suponer ya no solo una importante pérdida económica, si no poner en peligro las vidas de sus usuarios.

## 2.3. Buscando la calidad en la gestión de requisitos:

### 2.3.1. Buenas Prácticas:

#### 2.3.1.1. CMMI (“CMMI® para Desarrollo, Versión 1.3”):

CMMI son las siglas de *Capability Maturity Model Integration*. Según extraemos de la definición aportada en [4], CMMI es un modelo desarrollado por el *Software Engineering Institute* para la mejora de los procesos de las empresas de software y empleado inicialmente por el Departamento de Defensa que califica las compañías según su nivel de madurez. Por proceso se entiende un conjunto de fases sucesivas que llevan a la obtención de un resultado, y por nivel de madurez, el grado de calidad que alcanzan los procesos. Recientemente el modelo se encuentra en la versión 1.3.

Así pues, y en pocas palabras, CMMI establece una serie de buenas prácticas que las empresas deben cumplir para ser consideradas de un grado de madurez determinado a la hora de generar resultados. CMMI no indica cómo realizar esas buenas prácticas, sino que remarca los objetivos a conseguir mediante su implantación. Es labor de cada empresa adaptar esas prácticas acorde a sus posibilidades.

Existen tres tipos de CMMI, *CMMI for Acquisition*, para la mejora de procesos de contratación externa, la *CMMI for Services*, para el establecimiento y la entrega de servicios, y la que más nos interesa ***CMMI for Development***, para la mejora del desarrollo de productos software.

## Los niveles de madurez de CMMI

CMMI está representado, según [30], escalonadamente en 5 niveles de madurez, que califican a las empresas según la calidad de su proceso de producción. Estos niveles, además de por su correspondiente ordinal (del 1 al 5), se conocen por el adjetivo que los describe: Inicial, Gestionado, Definido, Cuantitativamente gestionado y el más alto de todos, Optimizado. Veamos brevísimamente en qué consiste cada uno de ellos.

Nivel 1 (Inicial): La organización no sigue conscientemente las prácticas especificadas en el modelo CMMI, pero de una forma u otra consigue que sus productos salgan al mercado. Supone una gran dependencia de los llamados héroes, profesionales sobresalientes que con su esfuerzo y habilidad personal sacan a la empresa del atolladero. La organización no genera conocimiento y la pérdida de estos héroes supone generalmente la pérdida de la capacidad acumulada.

Nivel 2 (Gestionado): Supone el cumplimiento, por parte de cada proyecto, de varias áreas de proceso de CMMI relacionadas con la gestión de requisitos, la planificación, la monitorización, la gestión de la configuración, el aseguramiento de la calidad, el acuerdo con los proveedores y la medición y análisis de los procesos. Se trata de asegurar que las buenas prácticas se mantengan independientemente de los vaivenes coyunturales que afecten a la organización. El conocimiento reside en la organización.

Nivel 3 (Definido): En este nivel, los procesos ya no sólo se definen de manera independiente para cada proyecto sino que toda la organización goza de unas pautas comunes. Se instaura una serie de procesos que, llevando la explicación al paradigma de la orientación a objetos, suponen la clase sobre la cual se instancian los proyectos. En resumidas cuentas, la empresa goza de una plantilla bien caracterizada que al ser aplicada a cada caso particular, genera un proyecto.

Nivel 4 (Cuantitativamente gestionado): Llegados a este punto, la organización ya no sólo gestiona los proyectos mediante procesos bien definidos, sino que además, se fijan objetivos tangibles que los procesos deben cumplir en lo relativo a su calidad, de manera que se analizan estadísticamente los procesos, propiciando una exactitud y predictibilidad, si se me permite el palabro, de la que no goza el nivel anterior.

Nivel 5 (Optimizado): En el nivel más alto de CMMI, la organización entera experimenta una optimización continua de los procesos a través de la innovación en los mismos y de las mejoras tecnológicas. Se modifican y reconducen los procesos en función de los defectos revelados durante el análisis estadístico.

## ¿Cómo se le reconoce oficialmente a una organización su nivel de madurez?

La evaluación, conocida como *Appraisal*, del grado de madurez de una empresa es llevada a cabo por un equipo que debe estar formado, según el método *SCAMPI* (*Standard CMMI Appraisal Method for Process Improvement*), por un evaluador oficial (*Lead Appraiser*) formado por el SEI y por un conjunto de personas que deben haber recibido el curso de introducción a CMMI y entre las que se puede encontrar, y de hecho es lo habitual, personal de la propia organización.

Se trata de una prueba exigente, que puede llegar a durar varias semanas, durante las cuales, entre otras cosas, se recogen y analizan las evidencias de cumplimiento de las áreas de proceso a evaluar y se entrevista a los involucrados en la organización. Al finalizar la *Appraisal*, el evaluador emite un informe que debe ser confirmado en Estados Unidos por el SEI.

## Conclusiones:

En resumen, CMMI, y más concretamente *CMMI for Development*, puede ser opinión un estupendo modelo para la mejora de procesos en una empresa, si es que esta organización se adecúa al modelo, es

decir: la organización persigue la mejora real antes que la certificación, está dispuesta a asumir el elevado coste que conlleva la implantación y evaluación y su tamaño y características evitan que CMMI suponga una elevada carga adicional a los proyectos.

### **2.3.1.1.1. CMMI - DESARROLLO DE REQUISITOS (Requirement Development RD)**

#### **Propósito**

El propósito del Desarrollo de Requisitos (RD) es educir, analizar y establecer los requisitos de cliente, de producto y de componente de producto. El desarrollo de requisitos está englobado en el área de nivel madurez 3.

#### **Notas introductorias**

Éste área de proceso describe tres tipos de requisitos: de cliente, de producto y de componente de producto. Tomados en conjunto, estos requisitos tratan las necesidades de las partes interesadas relevantes, incluyendo las necesidades pertinentes a las diferentes fases del ciclo de vida del producto (p. ej., criterios de pruebas de aceptación) y a los atributos del producto (p. ej., capacidad de respuesta, protección, fiabilidad, capacidad de mantenimiento). Los requisitos también tratan las restricciones causadas por la selección de soluciones de diseño (p. ej., integración de productos disponibles comercialmente (COTS), o uso de un patrón particular de arquitectura).

Todos los proyectos de desarrollo tienen requisitos. Los requisitos son la base para el diseño. El desarrollo de los requisitos incluye las siguientes actividades:

- Educación, análisis, validación y comunicación de las necesidades, las expectativas y las restricciones del cliente para obtener los requisitos priorizados de cliente que constituyen una comprensión de lo que satisfará a las partes interesadas.
- Recopilación y coordinación de las necesidades de las partes interesadas.
- Desarrollo de los requisitos del ciclo de vida del producto.
- Establecimiento de los requisitos funcionales de cliente y de los requisitos de los atributos de calidad.
- Establecimiento de los requisitos iniciales de producto y de componente de producto consistentes con los requisitos de cliente.

#### **LAS ÁREAS DE PROCESO**

Esta área de proceso trata todos los requisitos de cliente, y no sólo los requisitos a nivel de producto, ya que el cliente puede también proporcionar requisitos específicos de diseño.

Los requisitos de cliente se refinan más tarde en requisitos de producto y de componentes de producto. Además de los requisitos de cliente, los requisitos de producto y de componente de producto se derivan de las soluciones de diseño seleccionadas. En todas las áreas de proceso donde se utilizan los términos “producto” y “componente de producto”, su significado pretende también abarcar los servicios, los sistemas de servicio y sus componentes.

Los requisitos se identifican y se refinan a través de todas las fases del ciclo de vida del producto. Las decisiones de diseño, las acciones correctivas subsiguientes y la realimentación durante cada fase del ciclo de vida del producto se analizan para determinar el impacto sobre los requisitos derivados y asignados.

El área de proceso Desarrollo de Requisitos incluye tres metas específicas. La meta específica Desarrollar los requisitos de cliente trata la definición de un conjunto de requisitos de cliente para utilizarse en el desarrollo de los requisitos de producto. La meta específica Desarrollar los requisitos de producto trata la definición de un conjunto de requisitos de producto o de componente de producto para utilizarse en el diseño de productos y de componentes de producto. La meta específica Analizar y validar los requisitos trata el análisis de los requisitos de cliente, de producto y de componente de producto para definir, inferir y comprender los requisitos. Las prácticas específicas de la tercera meta específica están pensadas para ayudar a las prácticas específicas de las dos primeras metas específicas. Los procesos asociados al área de proceso Desarrollo de Requisitos y los procesos asociados con el área de proceso Solución Técnica pueden interactuar recursivamente unos con otros.

Los análisis se utilizan para comprender, definir y seleccionar los requisitos a todos los niveles a partir de alternativas competitivas. Estos análisis incluyen:

- Análisis de necesidades y de requisitos para cada fase del ciclo de vida de producto, incluyendo las necesidades de las partes interesadas relevantes, el entorno de operación y los factores que reflejan las expectativas y la satisfacción globales del cliente y del usuario final, tales como protección, seguridad y asequibilidad.
- Desarrollo de un concepto operacional.
- Definición de funcionalidad requerida y atributos de calidad.

Esta definición de funcionalidad requerida y de atributos de calidad describe qué tiene que hacer el producto. Esta definición puede incluir descripciones, descomposiciones y una división de las funciones (o en el análisis orientado a objetos lo que se ha referenciado como “servicios” o “métodos”) del producto. Además, la definición especifica las consideraciones o restricciones de diseño sobre cómo la funcionalidad requerida será realizada en el producto. Los atributos de calidad tratan cosas tales como: disponibilidad del producto; capacidad de mantenimiento; adaptabilidad; oportunidad, tasa de transferencia y capacidad de respuesta; fiabilidad; seguridad; y escalabilidad. Algunos atributos de calidad surgirán como significativos de la arquitectura y conducirán así el desarrollo de la arquitectura del producto.

Estos análisis ocurren recursivamente en capas sucesivas que contienen un mayor detalle de la arquitectura del producto hasta que se disponga del suficiente detalle para permitir realizar el diseño detallado, la adquisición y las pruebas del producto. Como resultado del análisis de requisitos y del concepto operacional (incluyendo funcionalidad, soporte, mantenimiento y retirada), la fabricación o el concepto de producción produce más requisitos derivados, incluyendo consideraciones como las siguientes:

- Restricciones de varios tipos.
- Limitaciones tecnológicas.
- Coste y parámetros de coste.
- Restricciones de tiempo y parámetros de calendario.
- Riesgos.
- Consideraciones de cuestiones implícitas, pero no declaradas explícitamente por el cliente o por el usuario final.
- Factores introducidos por consideraciones de negocio propias del desarrollador, por normativas y por leyes.

Una jerarquía de entidades lógicas (p. ej., funciones y subfunciones, clases y subclases de objetos; procesos; otras entidades arquitectónicas) se establece a través de la iteración con la evolución del concepto operacional. Los requisitos se refinan, se infieren y se asignan a estas entidades lógicas. Los requisitos y las entidades lógicas se asignan a los productos, a los componentes de producto, a las personas o a los procesos asociados. En el caso de desarrollo iterativo o incremental, los requisitos también se asignan a las iteraciones o incrementos. La involucración de las partes interesadas relevantes, tanto en el desarrollo como en el análisis de los requisitos, les proporciona visibilidad en la evolución de los requisitos. Esta actividad les asegura continuamente que los requisitos están siendo definidos apropiadamente.

Para las líneas de producto, los procesos de ingeniería (incluyendo el desarrollo de los requisitos) pueden aplicarse por lo menos a dos niveles en la organización. En un nivel de organización o de línea de producto, un “análisis de lo común y de la variación” se realiza para ayudar a educir, analizar y establecer activos base para su uso por los proyectos dentro de la línea de producto. En el nivel del proyecto, estos activos base son luego usados según el plan de producción de la línea de producto como parte de las actividades de ingeniería del proyecto.

En entornos ágiles, las necesidades y las ideas del cliente son iterativamente identificadas, elaboradas, analizadas y validadas. Los requisitos se documentan en formularios tales como: historias de usuario, escenarios, casos de uso, backlog del producto y los resultados de iteraciones (código en elaboración en el caso del software). Qué requisitos serán tratados en una iteración dada se determinan por una evaluación del riesgo y por las prioridades asociadas con los requisitos que se dejan en la backlog del producto. Qué detalles de los requisitos (y de otros artefactos) a documentar se determinan por la necesidad de coordinación (entre los miembros del equipo, los equipos y las iteraciones posteriores) y el riesgo de perder lo que se ha aprendido. Cuando el cliente está en el equipo, puede todavía existir una necesidad de separar la documentación de cliente y de producto para permitir que sean exploradas múltiples soluciones. Mientras que la solución surge, las responsabilidades de los requisitos derivados se asignan a los equipos apropiados.

## Resumen de metas y prácticas específicas

SG 1 Desarrollar los requisitos de cliente.

SP 1.1 Educir las necesidades.

SP 1.2 Trasformar las necesidades de las partes interesadas en requisitos de cliente.

SG 2 Desarrollar los requisitos de producto.

SP 2.1 Establecer los requisitos de producto y de componente de producto.

SP 2.2 Asignar los requisitos de componente de producto.

SP 2.3 Identificar los requisitos de interfaz.

SG 3 Analizar y validar los requisitos.

SP 3.1 Establecer los conceptos y los escenarios de operación.

SP 3.2 Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.

SP 3.3 Analizar los requisitos.

SP 3.4 Analizar los requisitos para conseguir un equilibrio.

SP 3.5 Validar los requisitos.

## Prácticas específicas por meta

### ***SG 1 Desarrollar los requisitos de cliente***

*Las necesidades, expectativas, restricciones e interfaces de las partes interesadas se recopilan y traducen en requisitos de cliente.*

Las necesidades de las partes interesadas (p. ej., clientes, usuarios finales, proveedores, desarrolladores, personal de pruebas, fabricantes, personal de soporte logístico) son la base para determinar los requisitos de cliente. Las necesidades, las expectativas, las restricciones, las interfaces, los conceptos operacionales y los conceptos de producto de las partes interesadas se analizan, ajustan, refinan y elaboran para traducirlos en un conjunto de requisitos de cliente.

Con frecuencia, las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas están pobremente identificadas o en conflicto. Puesto que las necesidades, las expectativas, las restricciones y las limitaciones de las partes interesadas deberían identificarse y comprenderse claramente, se utiliza un proceso iterativo durante toda la vida del proyecto para cumplir este objetivo. Para facilitar la interacción requerida, frecuentemente se involucra a un sustituto del usuario final o cliente para representar sus necesidades y ayudar a resolver conflictos. Las relaciones del cliente o el área de marketing de la organización, así como los miembros del equipo de desarrollo de disciplinas como ingeniería o soporte humano, pueden utilizarse como sustitutos. Las restricciones

medioambientales, legales y otras deberían considerarse al crear y resolver el conjunto de requisitos de cliente.

### ***SP 1.1 Educir las necesidades***

*Educir las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas para todas las fases del ciclo de vida del producto.*

Educir va más allá de la recopilación de requisitos mediante la identificación proactiva de requisitos adicionales no explícitamente proporcionados por los clientes. Los requisitos adicionales deberían tratar las distintas actividades del ciclo de vida del producto y sus impactos en el producto.

Algunos ejemplos de técnicas para educir las necesidades son:

- Demostraciones de tecnología.
- Grupos de trabajo de control de la interfaz.
- Grupos de trabajo de control técnico.
- Revisiones intermedias del proyecto.
- Cuestionarios, entrevistas y escenarios (operacional, soporte y desarrollo) obtenidos de los usuarios finales.
- Guías de soporte, desarrollo y de operación, y análisis de tareas de usuario final.
- Workshops con las partes interesadas para la reducción de los atributos de calidad.
- Prototipos y modelos.
- Tormenta de ideas.
- Despliegue de la función de calidad.
- Estudios de mercado.
- Pruebas beta.
- Extracción de fuentes como documentos, estándares o especificaciones.
- Observación de productos, entornos y patrones de flujo de trabajo existentes.
- Casos de uso.
- Historias de usuario.
- Pequeñas entregas incrementales “por rodajas” de la funcionalidad del producto.
- Análisis de casos de negocio.
- Ingeniería inversa (para productos heredados).
- Encuestas de satisfacción del cliente.

Algunos ejemplos de fuentes de requisitos que pueden no ser identificadas por el cliente son:

- Políticas de negocio.
- Estándares.
- Decisiones y principios de diseño de arquitectura previos.
- Requisitos de entorno de negocio (p. ej., laboratorios, pruebas y otras instalaciones, infraestructura de tecnología de información).
- Tecnología.
- Productos o componentes de producto heredados (reutilización de componentes de producto).
- Estatutos reguladores

#### *Ejemplo de productos de trabajo*

1. Resultados de las actividades de educación de requisitos.

#### *Subprácticas*

1. Comprometer a las partes interesadas relevantes usando métodos para educir las necesidades, las expectativas, las restricciones y las interfaces externas.

### ***SP 1.2 Transformar las necesidades de las partes interesadas en requisitos de cliente***

*Transformar las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas en requisitos de cliente priorizados.*

Se deberían consolidar las distintas entradas provenientes de las partes interesadas relevantes, se debería recuperar la información que falta y se deberían resolver los conflictos a medida que se desarrollan y priorizan los requisitos de cliente. Los requisitos de cliente pueden incluir las necesidades, las expectativas y las restricciones con respecto a la verificación y a la validación.

En algunas situaciones, el cliente proporciona un conjunto de requisitos al proyecto, o los requisitos existen como una salida de las actividades de un proyecto anterior. En estas situaciones, los requisitos de cliente podrían entrar en conflicto con las necesidades, expectativas, restricciones e interfaces de las partes interesadas relevantes y necesitarán transformarse en un conjunto reconocido de requisitos de cliente después de una resolución de conflictos apropiada.

Las partes interesadas relevantes que representan a todas las fases del ciclo de vida del producto deberían incluir tanto funciones del negocio, como funciones técnicas. De esta manera, los conceptos para todos los procesos del ciclo de vida relativo al producto se consideran concurrentemente con los conceptos para los productos. Los requisitos de cliente resultan tanto de decisiones informadas sobre el negocio, como de los efectos técnicos de sus requisitos.

#### *Ejemplos de productos de trabajo*

1. Requisitos de cliente priorizados.
2. Restricciones de cliente para llevar a cabo la verificación.
3. Restricciones de cliente para llevar a cabo la validación.

#### *Subprácticas*

1. Traducir las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas en requisitos documentados del cliente.
2. Establecer y mantener una priorización de los requisitos funcionales de cliente y de los atributos de calidad. Tener priorizados los requisitos de cliente ayuda a determinar el alcance del proyecto, de la iteración o del incremento. Esta priorización asegura que los requisitos funcionales y de los atributos de calidad críticos para el cliente y otras partes interesadas se tratan rápidamente.
3. Definir las restricciones para la verificación y la validación.

## **SG 2 Desarrollar los requisitos de producto**

*Los requisitos de cliente se refinan y elaboran para desarrollar los requisitos de producto y de componente de producto.*

Los requisitos de cliente se analizan conjuntamente con el desarrollo del concepto de operación para inferir conjuntos de requisitos más detallados y precisos llamados “requisitos de producto y de componente de producto”. Los requisitos de producto y de componente de producto tratan las necesidades asociadas con cada fase del ciclo de vida del producto. Los requisitos derivados surgen de restricciones; consideraciones de las cuestiones implícitas pero no declaradas explícitamente en la línea base de los requisitos de cliente; factores introducidos por la arquitectura seleccionada, ciclo de vida del producto y diseño; y por las consideraciones propias del negocio del desarrollador. Los requisitos se reexaminan con cada conjunto de requisitos sucesivo y de nivel más bajo y con la arquitectura, y se refina el concepto de producto preferido.

Los requisitos se asignan a las funciones del producto y a los componentes de producto, incluyendo objetos, personas y procesos. En el caso de desarrollo iterativo o incremental, los requisitos también se asignan a las iteraciones o a los incrementos en base a las prioridades del cliente, cuestiones de tecnología y objetivos del proyecto. Se documenta la trazabilidad de los requisitos a las funciones, objetos, pruebas, cuestiones u otras entidades. Los requisitos asignados y las funciones (u otras entidades lógicas) son la base para la síntesis de la solución técnica; sin embargo, mientras se define la arquitectura, sirve como base última para dirigir la asignación de requisitos a la solución.

A medida que se desarrollan los componentes internos, se definen interfaces adicionales y se establecen los requisitos de la interfaz.



**SP 2.1 Establecer los requisitos de producto y de componente de producto**

*Establecer y mantener los requisitos de producto y de componente de producto, basados en los requisitos de cliente.*

Los requisitos funcionales y los atributos de calidad del cliente pueden expresarse en términos del cliente y pueden ser descripciones no técnicas.

Los requisitos de producto son la expresión de estos requisitos en términos técnicos que pueden utilizarse para las decisiones de diseño.

Un ejemplo de esta traducción se encuentra en la primera House of Quality Function Deployment, que traduce los deseos del cliente en parámetros técnicos. Por ejemplo, “puerta sólida de resonancia” podría corresponder a tamaño, peso, ajuste, amortiguación y frecuencias de resonancia.

Los requisitos de producto y de componente de producto tratan la satisfacción del cliente, el negocio y los objetivos del proyecto y sus atributos asociados, tales como eficacia y asequibilidad.

Los requisitos derivados también tratan las necesidades de otras fases del ciclo de vida (p. ej., producción, operaciones, retirada) en la medida que son compatibles con los objetivos de negocio.

La modificación de los requisitos debida a cambios de requisitos aprobados se cubre por el aspecto “mantener” de esta práctica específica; mientras que la gestión de los cambios a los requisitos se cubre por el área de proceso Gestión de Requisitos.

**Ejemplos de productos de trabajo**

1. Requisitos derivados.
2. Requisitos de producto.
3. Requisitos de componente de producto.
4. Requisitos de arquitectura, que especifican o restringen las relaciones entre componentes de producto.

**Subprácticas**

1. Desarrollar los requisitos en los términos técnicos necesarios para el diseño del producto y de los componentes de producto.

2. Inferir los requisitos resultantes de las decisiones de diseño. La selección de una tecnología trae consigo requisitos adicionales. Por ejemplo, el uso de electrónica requiere requisitos tecnológicos específicos adicionales, tales como límites de interferencia electromagnética.

Las decisiones de arquitectura, tales como la selección de patrones de la arquitectura, introducen requisitos inferidos adicionales para los componentes de producto. Por ejemplo, el patrón de capas restringirá dependencias entre ciertos componentes de producto.

3. Desarrollar los requisitos de arquitectura capturando los atributos críticos de calidad y las medidas de atributos de calidad necesarios para establecer la arquitectura y el diseño del producto.

Algunos ejemplos de medidas de los atributos de calidad son:

- Respuesta dentro de 1 segundo.
  - Disponibilidad del sistema en un 99% del tiempo.
  - Implementar un cambio con no más de una semana de esfuerzo del personal.
4. Establecer y mantener las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos.

Las relaciones entre los requisitos pueden ayudar en la evaluación del impacto de los cambios.

**SP 2.2 Asignar los requisitos de componente de producto**

*Asignar los requisitos para cada componente de producto.*

La arquitectura del producto proporciona la base para asignar los requisitos de producto a los componentes de producto. Los requisitos para los componentes de producto de la solución definida incluyen la asignación de la prestación del producto; las restricciones de diseño; y el ajuste, la forma y la función para cumplir los requisitos y facilitar la producción. En los casos donde un requisito de nivel más alto especifique un atributo de calidad que sea responsabilidad de más de un componente de producto, el atributo de calidad puede a veces dividirse para la asignación única a cada componente de producto

como un requisito inferido, sin embargo, otras veces el requisito compartido debería, en su lugar, estar asignado directamente a la arquitectura.

Por ejemplo, la asignación de requisitos compartidos a la arquitectura podría describir cómo se presupuesta un requisito de rendimiento (p. ej, en capacidad de respuesta) entre los componentes, de forma que se contabilice de inicio a fin la forma de realizar el requisito. Este concepto de requisitos compartidos puede extenderse a otros atributos de calidad de arquitectura significativos (p. ej, seguridad, fiabilidad).

#### *Ejemplos de productos de trabajo*

1. Hojas de asignación de requisitos.
2. Asignaciones provisionales de requisitos.
3. Restricciones de diseño.
4. Requisitos inferidos.
5. Relaciones entre requisitos inferidos.

#### *Subprácticas*

1. Asignar los requisitos a las funciones.
2. Asignar los requisitos a los componentes de producto y a la arquitectura.
3. Asignar las restricciones de diseño a componentes de producto y a la arquitectura.
4. Asignar requisitos a las entregas incrementales.
5. Documentar las relaciones entre requisitos asignados.

Las relaciones incluyen las dependencias en que un cambio en un requisito puede afectar a otros requisitos.

### **SP 2.3 Identificar los requisitos de interfaz**

*Identificar los requisitos de interfaz.*

Se identifican las interfaces entre las funciones (o entre objetos u otras entidades lógicas). Las interfaces pueden conducir el desarrollo de soluciones alternativas descritas en el área de proceso Solución Técnica.

Se definen los requisitos de interfaz entre los productos o componentes de producto identificados en la arquitectura del producto. Estos requisitos se controlan como parte de la integración del producto y del componente de producto y son una parte integral de la definición de la arquitectura.

#### *Ejemplo de productos de trabajo*

1. Requisitos de interfaz.

#### *Subprácticas*

1. Identificar las interfaces tanto externas como internas al producto (p. ej., entre las particiones funcionales u objetos).

A medida que progresa el diseño, la arquitectura del producto será alterada por los procesos de la solución técnica, creando nuevas interfaces entre los componentes de producto y los componentes externos al producto.

Deberían identificarse también las interfaces con los procesos del ciclo de vida relativos al producto.

Algunos ejemplos de estas interfaces son las interfaces con el equipo de pruebas, con los sistemas de transporte, con los sistemas de soporte y con las instalaciones de fabricación.

2. Desarrollar los requisitos para las interfaces identificadas.

Los requisitos para las interfaces se definen en términos tales como el origen, el destino, el estímulo, las características de los datos para el software, y las características eléctricas y mecánicas para el hardware.

### **SG 3 Analizar y validar requisitos**

*Los requisitos se analizan y validan.*

Las prácticas específicas de la meta específica *Analizar y validar requisitos* dan soporte al desarrollo de los requisitos, tanto en la meta específica *Desarrollar los requisitos de cliente* como en la meta específica *Desarrollar los requisitos de producto*. Las prácticas específicas asociadas a esta meta específica cubren el análisis y la validación de los requisitos con respecto al entorno previsto del usuario final.

Los análisis se realizan para determinar qué impacto tendrá el entorno de operación previsto sobre la capacidad para satisfacer las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas. Deberían tenerse en cuenta consideraciones como la viabilidad, las necesidades de la misión, las restricciones de coste, el tamaño del mercado potencial y la estrategia de la adquisición, dependiendo del contexto del producto. Los atributos de calidad de la arquitectura significativos se identifican en base a la misión y a los factores del negocio. También se establece una definición de la funcionalidad requerida y de los atributos de calidad. Se consideran todos los modos de uso especificados para el producto.

Los objetivos de los análisis son determinar los requisitos candidatos para los conceptos del producto que satisfarán las necesidades, las expectativas y las restricciones de las partes interesadas, y entonces traducir estos conceptos en requisitos. Paralelamente a esta actividad, los parámetros que se utilizarán para evaluar la eficacia del producto se determinan en base a las entradas del cliente y al concepto preliminar del producto.

Los requisitos se validan para incrementar la probabilidad de que el producto resultante funcione según lo previsto en el entorno de uso.

### ***SP 3.1 Establecer los conceptos y los escenarios de operación***

*Establecer y mantener los conceptos y los escenarios de operación asociados.*

Un escenario es normalmente una secuencia de eventos que podrían ocurrir en el desarrollo, uso o soporte del producto, el cual se usa para hacer explícitas algunas de las necesidades funcionales o de los atributos de calidad de las partes interesadas. Por el contrario, un concepto operacional para un producto depende generalmente tanto de la solución de diseño como del escenario. Por ejemplo, el concepto operacional para un producto de comunicaciones basado en satélites es completamente diferente de uno basado en comunicaciones por tierra. Puesto que las soluciones alternativas generalmente no han sido definidas cuando se preparan los conceptos de operación iniciales, las soluciones conceptuales se desarrollan para su uso cuando se analizan los requisitos. Los conceptos operacionales se refinan a medida que se toman las decisiones de la solución y se desarrollan los requisitos detallados de más bajo nivel.

Al igual que una decisión de diseño para un producto puede llegar a ser un requisito para un componente de producto, el concepto de operación puede llegar a ser los escenarios (requisitos) para los componentes de producto. Los conceptos y los escenarios de operación evolucionan para facilitar la selección de soluciones del componente de producto que, cuando se implementan, satisfarán el uso previsto del producto o facilitarán su desarrollo y soporte. Los conceptos y los escenarios de operación documentan la interacción de los componentes de producto con el entorno, con los usuarios finales y con otros componentes de producto, independientemente de la disciplina de ingeniería. Estos conceptos se deberían documentar para todos los modos y los estados dentro de las operaciones, del desarrollo del producto, del despliegue, de la entrega, del soporte (incluyendo el mantenimiento y soporte), de la formación y de la retirada.

Los escenarios pueden desarrollarse para tratar secuencias de operación, de soporte, de desarrollo o de otros eventos.

#### *Ejemplos de productos de trabajo*

1. Concepto operacional.
2. Desarrollo, instalación, operación, mantenimiento y conceptos de soporte del producto o componente de producto.
3. Conceptos de retirada.
4. Casos de uso.
5. Escenarios de cronología.
6. Nuevos requisitos.

#### *Subprácticas*

1. Desarrollar los conceptos y los escenarios de operación que incluyan operaciones, instalación, desarrollo, mantenimiento, soporte y retirada según sea apropiado.

Identifique y desarrolle los escenarios, consistentes con el nivel de detalle de las necesidades, las expectativas y las restricciones de las partes interesadas en las cuales se espera que funcione el producto o componente de producto propuesto.

Aumente los escenarios con las consideraciones de los atributos de calidad para las funciones (u otras entidades lógicas) descritas en los escenarios.

2. Definir el entorno en el que funcionará el producto o componente de producto, incluyendo los límites y las restricciones.

3. Revisar los conceptos y los escenarios de operación para refinar y descubrir requisitos.

El concepto y el desarrollo del escenario de operación es un proceso iterativo. Para asegurar que éstos están de acuerdo con los requisitos, deberían mantenerse revisiones periódicamente. La revisión puede ser guiada.

4. Desarrollar un concepto de operación detallado, a medida que se seleccionan los productos y los componentes de producto, que define la interacción del producto, del usuario final y del entorno, y que satisfaga las necesidades operativas, de mantenimiento, de soporte y de retirada.

### ***SP 3.2 Establecer una definición de la funcionalidad y de los atributos de calidad requeridos***

*Establecer y mantener una definición de funcionalidad y de los atributos de calidad requeridos.*

Una aproximación para definir la funcionalidad y los atributos de calidad requeridos es analizar escenarios utilizando lo que llamamos “análisis funcional” para describir lo que se pretende que haga el producto.

Esta descripción funcional puede incluir acciones, secuencia, entradas, salidas u otra información que comunique la forma en la que se utilizará el producto. La descripción resultante de las funciones, agrupaciones lógicas de funciones y su asociación con los requisitos se denomina como arquitectura funcional (véanse las definiciones de

“análisis funcional” y “arquitectura funcional” en el glosario).

Estas aproximaciones han evolucionado en años recientes a través de la introducción de lenguajes, métodos y herramientas de descripción de la arquitectura para tratar y caracterizar de forma más completa los atributos de calidad, permitiendo una especificación más rica de las restricciones sobre cómo la funcionalidad definida se realizará en el producto (p.ej., multidimensional), y facilitando análisis adicionales de los requisitos y las soluciones técnicas. Algunos atributos de calidad que surgirán serán significativos para la arquitectura y así conducirán el desarrollo de la arquitectura del producto. Estos atributos de calidad reflejan a menudo temas transversales que no se pudieron asignar a los elementos del nivel más bajo de una solución. Una comprensión clara de los atributos de calidad y de su importancia en base a la misión o necesidades del negocio es una entrada esencial al proceso de diseño.

#### *Ejemplos de productos de trabajo*

1. Definición de funcionalidad y de atributos de calidad requeridos.
2. Arquitectura funcional.
3. Diagramas de actividad y casos de uso.
4. Análisis orientado a objetos con servicios o métodos identificados.
5. Requisitos de los atributos de calidad significativos para la arquitectura.

#### *Subprácticas*

1. Determinar la misión clave y los factores de negocio.
2. Identificar la funcionalidad y los atributos de calidad deseados.

La funcionalidad y los atributos de calidad pueden ser identificados y definidos con las partes interesadas relevantes a través de un análisis de diversos escenarios según lo descrito en la práctica específica anterior.

3. Determinar los atributos de calidad significativos para la arquitectura en base a la misión clave y los factores del negocio.
4. Analizar y cuantificar la funcionalidad requerida por los usuarios finales.  
Este análisis puede implicar considerar la secuencia de funciones críticas de tiempo.
5. Analizar los requisitos para identificar las particiones lógicas o funcionales (p. ej., subfunciones).
6. Dividir los requisitos en grupos, en base a los criterios establecidos (p. ej., funcionalidad similar, requisitos similares de atributos de calidad, acoplamiento) para facilitar y enfocar el análisis de requisitos.
7. Asignar los requisitos de cliente a las particiones funcionales, objetos, personal o elementos de soporte para dar apoyo a la síntesis de las soluciones.

8. Asignar los requisitos a las funciones y a las subfunciones (u otras entidades lógicas).

### **SP 3.3 Analizar los requisitos**

*Analizar los requisitos para asegurarse que son necesarios y suficientes.*

Teniendo en cuenta el concepto y los escenarios de operación, los requisitos para el nivel uno de la jerarquía del producto se analizan para determinar si son necesarios y suficientes para cumplir con los objetivos de niveles más altos de la jerarquía del producto. Los requisitos analizados, posteriormente, proporcionan la base para requisitos más detallados y precisos en los niveles más bajos de la jerarquía del producto.

A medida que se definen los requisitos, debería comprenderse su relación con los requisitos de nivel más alto y la definición de más alto nivel de la funcionalidad y los atributos de calidad requeridos. También, se determinan los requisitos clave que se utilizarán para realizar el seguimiento del progreso. Por ejemplo, el peso de un producto o el tamaño de un producto software pueden monitorizarse durante el desarrollo en base al riesgo o la criticidad que tienen para el cliente.

#### *Ejemplos de productos de trabajo*

1. Informes de defectos de los requisitos.
2. Cambios propuestos a los requisitos para resolver defectos.
3. Requisitos clave.
4. Medidas de rendimiento técnico.

#### *Subprácticas*

1. Analizar las necesidades, las expectativas, las restricciones y las interfaces externas de las partes interesadas para organizarlas en temas relacionados y eliminar conflictos.
2. Analizar los requisitos para determinar si satisfacen los objetivos de los requisitos de nivel más alto.
3. Analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables. Mientras que el diseño determina la viabilidad de una solución particular, esta subpráctica trata de conocer qué requisitos afectan a la viabilidad.
4. Identificar los requisitos clave que tienen una fuerte influencia en el coste, el calendario, el rendimiento o el riesgo.
5. Identificar las medidas técnicas de rendimiento que serán seguidas durante el esfuerzo de desarrollo.
6. Analizar los conceptos y los escenarios de operación para refinar las necesidades, las restricciones y las interfaces del cliente, y para Inferir nuevos requisitos.

Este análisis puede dar lugar a conceptos y a escenarios de operación más detallados, así como al soporte de la inferencia de nuevos requisitos.

### **SP 3.4 Analizar los requisitos para conseguir un equilibrio**

*Analizar los requisitos para equilibrar las necesidades y las restricciones de las partes interesadas.*

Las necesidades y las restricciones de las partes interesadas pueden tratar temas como coste, calendario, rendimiento del proyecto o producto, funcionalidad, prioridades, componentes reutilizables, facilidad de mantenimiento o riesgo.

#### *Ejemplo de productos de trabajo típicos*

1. Evaluación de los riesgos relativos a los requisitos.

#### *Subprácticas*

1. Usar modelos, simulaciones y prototipos probados para analizar el equilibrio entre las necesidades y las restricciones de las partes interesadas. Los resultados de los análisis pueden usarse para reducir el coste del producto y el riesgo del desarrollo del producto.
2. Realizar una evaluación de riesgos sobre los requisitos y la definición de funcionalidad y atributos de calidad requeridos.
3. Examinar los conceptos del ciclo de vida del producto en cuanto a los impactos de los requisitos en los riesgos.

4. Evaluar el impacto de los requisitos de los atributos de calidad significativos para la arquitectura en el producto y en los costes y riesgos del desarrollo del producto.

Cuando el impacto de los requisitos en costes y riesgos parece superar el beneficio percibido, debería consultarse a las partes interesadas relevantes para determinar qué cambios pueden ser necesarios.

Como ejemplo, un requisito tiempo-respuesta realmente ajustado o un requisito de alta-disponibilidad, podría demostrar un coste alto de implementación. Quizás el requisito podría relajarse una vez que se entienden los impactos (p. ej., en coste).

### **SP 3.5 Validar los requisitos**

*Validar los requisitos para asegurar que el producto resultante funcione según lo previsto en el entorno del usuario final.*

La validación de los requisitos se ejecuta en etapas tempranas del desarrollo con los usuarios finales para ganar confianza en que los requisitos sean capaces de guiar un desarrollo que dé como resultado una validación final de éxito. Esta actividad debería integrarse con las actividades de gestión de riesgos. Las organizaciones maduras normalmente realizarán la validación de los requisitos de una manera más sofisticada, usando múltiples técnicas y ampliarán la base de la validación para incluir otras necesidades y expectativas de las partes interesadas.

Algunos ejemplos de técnicas usadas para la validación de los requisitos son:

- Análisis.
- Simulaciones.
- Prototipos.
- Demostraciones.

#### *Ejemplo de productos de trabajo*

1. Registro de los métodos y resultados del análisis.

#### *Subprácticas*

1. Analizar los requisitos para determinar el riesgo de que el producto resultante no funcione apropiadamente en el entorno de uso previsto.
2. Explorar la adecuación y la completitud de los requisitos desarrollando representaciones del producto (p. ej., prototipos, simulaciones, modelos, escenarios, guiones gráficos) y obteniendo realimentación sobre ellos de las partes interesadas relevantes.
3. Evaluar el diseño a medida que madure en el contexto del entorno de validación de los requisitos para identificar las cuestiones de validación, y para exponer necesidades y requisitos de cliente sin especificar.

## **2.3.1.1.2. GESTIÓN DE REQUISITOS (Requirement Management REQM)**

### **Propósito**

Como se describe en [9] el propósito de la Gestión de Requisitos (REQM) es gestionar los requisitos de los productos y los componentes de producto del proyecto, y asegurar la alineación entre esos requisitos, y los planes y los productos de trabajo del proyecto. La gestión de requisitos está englobada en el área de nivel madurez.

### **Notas introductorias**

Los procesos de gestión de requisitos gestionan todos los requisitos recibidos o generados por el proyecto, incluyendo tanto los requisitos técnicos como los no técnicos, así como los requisitos impuestos al proyecto por la organización.

En particular, si se implementa el área de proceso Desarrollo de Requisitos, sus procesos generarán requisitos de producto y de componente de producto que también serán gestionados por los procesos de gestión de requisitos.

En todas las áreas de proceso, cuando se utilizan los términos “producto” y “componente de producto”, sus significados previstos también incluyen los servicios, los sistemas de servicio y sus componentes. Cuando las áreas de proceso Gestión de Requisitos, Desarrollo de Requisitos y Solución Técnica están implementadas, sus procesos asociados pueden estar estrechamente relacionados y realizarse de manera concurrente.

El proyecto realiza los pasos apropiados para asegurar que el conjunto de requisitos aprobados se gestiona para dar soporte a las necesidades de planificación y de ejecución del proyecto. Cuando un proyecto recibe requisitos de un proveedor de requisitos aprobado, éstos se revisan con dicho proveedor para resolver las cuestiones y para prevenir malentendidos antes de que los requisitos se incorporen en los planes del proyecto. Una vez que el proveedor y el receptor de los requisitos alcanzan un acuerdo, se obtiene un compromiso sobre los requisitos por parte de los participantes en el proyecto. El proyecto gestiona los cambios a los requisitos a medida que evolucionan e identifica inconsistencias que ocurren entre los planes, los productos de trabajo y los requisitos.

Una parte de la gestión de requisitos es documentar los cambios a los requisitos y su análisis razonado, y mantener la trazabilidad bidireccional entre los requisitos fuente, todos los requisitos de producto y de componente de producto, y otros productos de trabajo especificados (véase la definición de “trazabilidad bidireccional” en el glosario).

Todos los proyectos tienen requisitos. En el caso de actividades de mantenimiento, los cambios se basan en los cambios a los requisitos, al diseño o a la implementación existente. En proyectos que entregan incrementos de la capacidad del producto, los cambios también se pueden deber a la evolución de las necesidades del cliente, a la madurez u obsolescencia de la tecnología y a la evolución de los estándares. En ambos casos, los cambios a los requisitos, si existen, podrían documentarse en peticiones de cambio del cliente o de los usuarios finales, o podrían tomar la forma de nuevos requisitos recibidos del proceso de desarrollo de requisitos. Independientemente de su fuente o forma, las actividades que son dirigidas por cambios a los requisitos se gestionan consecuentemente.

En entornos ágiles, los requisitos se comunican y se siguen a través de mecanismos, tales como, backlog de producto, tarjetas de historia y maquetas de pantallas. Los compromisos a los requisitos se hacen bien colectivamente por el equipo o por un líder de equipo autorizado.

Las asignaciones del trabajo se ajustan regularmente (p.ej., diaria, semanalmente) en base al progreso obtenido y por una mejor comprensión de los requisitos y de la solución. La trazabilidad y la consistencia entre los requisitos y los productos de trabajo se tratan a través de los mecanismos ya mencionados, así como durante las actividades de comienzo de iteración o de fin de iteración, tales como, “reuniones de retrospectiva” y “días demo”.

## Resumen de metas y prácticas específicas

SG 1 Gestionar los requisitos.

SP 1.1 Comprender los requisitos.

SP 1.2 Obtener el compromiso sobre los requisitos.

SP 1.3 Gestionar los cambios a los requisitos.

SP 1.4 Mantener la trazabilidad bidireccional de los requisitos.

SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos.

### Prácticas específicas por meta

#### **SG 1 Gestionar los requisitos**

*Los requisitos se gestionan y las inconsistencias con los planes y productos de trabajo del proyecto se identifican.*

El proyecto mantiene un conjunto actual y aprobado de requisitos durante la vida del proyecto haciendo lo siguiente:

- Gestionando todos los cambios a los requisitos.
- Manteniendo las relaciones entre los requisitos, los planes del proyecto y los productos de trabajo.
- Asegurando la alineación entre los requisitos, los planes del proyecto y los productos de trabajo.



- Tomando acciones correctivas.

### **SP 1.1 Comprender los requisitos**

*Desarrollar una comprensión del significado de los requisitos con los proveedores de los requisitos.*

A medida que madura el proyecto y se derivan los requisitos, todas las actividades o disciplinas recibirán requisitos. Para evitar el flujo continuo de requisitos, se establecen criterios para designar los canales apropiados o las fuentes oficiales desde las que se reciben los requisitos.

Aquellos que reciben los requisitos, los analizan con el proveedor para asegurar que se alcanza una comprensión compatible y compartida del significado de los requisitos. El resultado de estos análisis y diálogos es un conjunto de requisitos aprobados.

#### *Ejemplos de productos de trabajo*

1. Listas de criterios para distinguir a los proveedores apropiados de requisitos.
2. Criterios para la evaluación y la aceptación de los requisitos.
3. Resultados del análisis frente a los criterios.
4. Un conjunto de requisitos aprobados.

#### *Subprácticas*

1. Establecer criterios para distinguir a los proveedores apropiados de requisitos.
  2. Establecer criterios objetivos para la evaluación y aceptación de los requisitos.
- La falta de criterios de evaluación y de aceptación da lugar a menudo a una verificación inadecuada, un retrabajo costoso o el rechazo del cliente.

Algunos ejemplos de criterios de evaluación y de aceptación son:

- Claramente y correctamente establecidos.
- Completos.
- Consistentes unos con otros.
- Identificados de forma única.
- Consistentes con el enfoque de la arquitectura y con las prioridades de los atributos de calidad.
- Apropiados para implementar.
- Verificables (es decir, que se pueden probar).
- Trazables.
- Alcanzables.
- Vinculados al valor de negocio.
- Identificados como una prioridad para el cliente.

3. Analizar los requisitos para asegurar que se cumplen los criterios establecidos.
4. Alcanzar una comprensión de los requisitos con los proveedores de requisitos para que los participantes del proyecto puedan comprometerse con ellos.

### **SP 1.2 Obtener el compromiso sobre los requisitos**

*Obtener el compromiso de los participantes del proyecto sobre los requisitos.*

La práctica específica anterior se ocupó de alcanzar una comprensión con los proveedores de los requisitos. Esta práctica específica se ocupa de los acuerdos y compromisos entre aquellos que llevan a cabo las actividades necesarias para implementar los requisitos. Los requisitos evolucionan a lo largo del proyecto. A medida que los requisitos evolucionan, esta práctica específica asegura que los participantes del proyecto se comprometen con los requisitos actuales y aprobados, y con los cambios resultantes en los planes, actividades y productos de trabajo del proyecto.

#### *Ejemplos de productos de trabajo*

1. Evaluaciones del impacto de los requisitos.
2. Compromisos documentados de los requisitos y de sus cambios.



#### *Subprácticas*

1. Evaluar el impacto de los requisitos sobre los compromisos existentes.

El impacto sobre los participantes del proyecto se debería evaluar cuando cambian los requisitos o al principio de un nuevo requisito.

2. Negociar y registrar los compromisos.

Los cambios a los compromisos existentes se deberían negociar antes de que los participantes del proyecto se comprometan con un nuevo requisito o un cambio a un requisito.

### **SP 1.3 Gestionar los cambios a los requisitos**

*Gestionar los cambios a los requisitos a medida que evolucionan durante el proyecto.*

Los requisitos cambian por diversas razones. A medida que cambian las necesidades y avanza el trabajo, es posible que se tengan que hacer cambios a los requisitos existentes. Es esencial gestionar estas adiciones y cambios, eficiente y eficazmente. Para analizar con eficacia el impacto de los cambios, es necesario que se conozca la fuente de cada requisito y que esté documentado el análisis razonado de cualquier cambio. El proyecto puede querer seguir medidas apropiadas de volatilidad de los requisitos para juzgar si es necesario un enfoque nuevo o modificado para el control de cambios.

#### *Ejemplos de productos de trabajo*

1. Petición de cambio de requisitos.
2. Informes de impacto del cambio de requisitos.
3. Estado de los requisitos.
4. Base de datos de requisitos.

#### *Subprácticas*

1. Documentar todos los requisitos y los cambios a los requisitos que se reciben o generan por el proyecto.
2. Mantener una historia de cambios de los requisitos, incluyendo el análisis razonado de los cambios. Mantener la historia de cambios ayuda a seguir la volatilidad de los requisitos.
3. Evaluar el impacto de los cambios de requisitos desde el punto de vista de las partes interesadas relevantes. Los cambios a los requisitos que afectan a la arquitectura del producto pueden afectar a muchas partes interesadas.
4. Poner a disposición del proyecto los requisitos y los datos de los cambios.

### **SP 1.4 Mantener la trazabilidad bidireccional de los requisitos**

*Mantener la trazabilidad bidireccional entre los requisitos y los productos de trabajo.*

La intención de esta práctica específica es mantener la trazabilidad bidireccional de los requisitos (véase la definición de “trazabilidad bidireccional” en el glosario). Cuando se gestionan bien los requisitos, se puede establecer la trazabilidad desde un requisito fuente hasta sus requisitos de más bajo nivel y desde estos requisitos de más bajo nivel de vuelta hasta sus requisitos fuente. Esta trazabilidad bidireccional ayuda a determinar si todos los requisitos fuente se han tratado totalmente y si todos los requisitos de nivel más bajo pueden trazarse hacia una fuente válida.

La trazabilidad de los requisitos también cubre las relaciones a otras entidades, tales como productos de trabajo intermedios y finales, cambios en la documentación de diseño y planes de pruebas. La trazabilidad puede cubrir relaciones horizontales, tales como relaciones entre interfaces, así como relaciones verticales. La trazabilidad es particularmente necesaria al evaluar el impacto de los cambios de los requisitos sobre las actividades del proyecto y los productos de trabajo.

Algunos ejemplos de aspectos de trazabilidad a considerar son:

- Alcance de la trazabilidad: los límites dentro de los cuales es necesaria la trazabilidad.
- Definición de trazabilidad: los elementos que necesitan relaciones lógicas.
- Tipos de trazabilidad: cuándo es necesaria la trazabilidad horizontal y la vertical.

La trazabilidad bidireccional no siempre está automatizada. Ésta se puede hacer manualmente utilizando hojas de cálculo, bases de datos u otras herramientas comunes.

*Ejemplos de productos de trabajo*

1. Matriz de trazabilidad de los requisitos.
2. Sistema de seguimiento de los requisitos.

*Subprácticas*

1. Mantener la trazabilidad de los requisitos para asegurar que la fuente de los requisitos de nivel más bajo (es decir, inferidos) está documentada.
2. Mantener la trazabilidad de los requisitos desde un requisito a sus requisitos inferidos y a la asignación a los productos de trabajo.

Los productos de trabajo para los cuales la trazabilidad se puede mantener incluyen la arquitectura, los componentes de producto, las iteraciones de desarrollo (o incrementos), las funciones, las interfaces, los objetos, las personas, los procesos y otros productos de trabajo.

3. Generar una matriz de trazabilidad de requisitos.

**SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos**

*Asegurar que los planes del proyecto y los productos de trabajo permanecen alineados con los requisitos.*

Esta práctica específica encuentra las inconsistencias entre los requisitos, los planes del proyecto y los productos de trabajo, e inicia acciones correctivas para resolverlas.

*Ejemplos de productos de trabajo*

1. Documentación de inconsistencias entre los requisitos y los planes del proyecto y los productos de trabajo, incluyendo fuentes y condiciones.
2. Acciones correctivas.

*Subprácticas*

1. Revisar los planes del proyecto, las actividades y los productos de trabajo en cuanto a la consistencia con los requisitos y los cambios realizados sobre ellos.
2. Identificar la fuente de la inconsistencia (si existe).
3. Identificar cualquier cambio que se debería realizar a los planes y a los productos de trabajo resultantes de los cambios a la línea base de requisitos.
4. Iniciar cualquier acción correctiva necesaria.

**2.3.1.1.3. Conclusiones:**

Siendo interesantes las recomendaciones de CMMI en cuanto a los requisitos, es necesario establecer los procesos que se van a llevar a cabo para alcanzar los objetivos que proponen las áreas de proceso. Como recomendaciones que son, no proponen una taxonomía de requisitos, los atributos que estos deben contemplar, ni el ciclo de vida que deben seguir. De la misma forma, la validación y verificación, como actividades fundamentales en el desarrollo y gestión de los requisitos, tampoco son expuestas con detalle.

CMMI no es una metodología, sino un conjunto de buenas prácticas.

En la figura 8 se resumen las consideraciones que CMMI propone acerca del tratamiento de los requisitos en las dos áreas de proceso que describe.

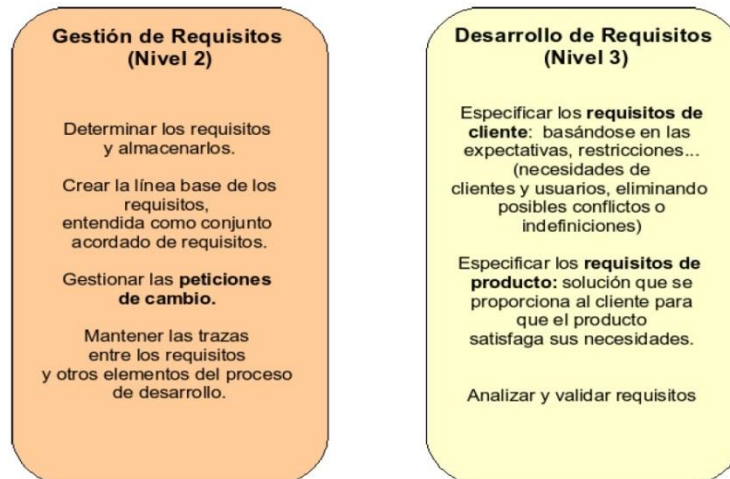


Figura 8: Resumen tratamiento Requisitos en CMMI v1.3

### 2.3.2. ITIL:

La *Biblioteca de Infraestructura de Tecnologías de Información*, frecuentemente abreviada ITIL (del inglés *Information Technology Infrastructure Library*), es un conjunto de conceptos y prácticas para la gestión de servicios de tecnologías de la información, el desarrollo de tecnologías de la información y las operaciones relacionadas con la misma en general según la descripción disponible en [5]. ITIL da descripciones detalladas de un extenso conjunto de procedimientos de gestión ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las operaciones de TI. Estos procedimientos son independientes del proveedor y han sido desarrollados para servir como guía que abarque toda infraestructura, desarrollo y operaciones de TI. Hoy en día ITIL es el estándar de facto en la Gestión del Servicio.

#### 2.3.2.1. **Conceptos Generales de ITIL**

ITIL fue creado a finales de la década de los ochenta por la *Central Computer and Telecommunication Agency (CCTA)* del Reino Unido. A día de hoy está regulado y patentado por el *Ministerio de Comercio (OCG)* pero externalizado también a las empresas privadas por el *IT SMF (Information Technology Service Management Forum)*. Dicha organización centra sus actividades, para el ámbito privado, en la difusión de ITIL.

En el 2000 fue publicada la versión 2 de ITIL, teniendo en el 2007 ITIL V3.

ITIL se define como una biblioteca que describe las buenas prácticas en la Gestión de Servicios. Al igual que CMMI, no es una metodología ya que no definen el cómo sino que atienden al qué.

El valor de ITIL radica en enfocar los servicios tecnológicos como activos de la organización, formando parte fundamental del plan estratégico de la organización, es decir, los servicios que ofrece cualquier organización a sus usuarios y/o clientes, se basan en una infraestructura tecnológica.

ITIL v2 proporciona estructura en los procesos, mientras que la evolución a v3 proporciona mejora continua en los servicios. Si se piensa en una implantación de ITIL en una organización, primero habría que comenzar por implantar los procesos de la v2, para poder pasar a la mejora continua propuesta por la v3.

Esta última versión integra los procesos y funciones de la ISO 20000, ya que es el estándar de la Gestión del Servicio de TI. Dichos procesos son cubiertos dentro de los procesos definidos en ITIL en la Estrategia del Servicio y Diseño del Servicio, que profundizaremos más adelante.

Es decir, aumentando el grado de abstracción y teniendo en cuenta que ITIL no es una metodología, tendríamos en la base los procedimientos y procesos implantados en la organización, guiados por las

buenas prácticas de ITIL. Teniendo en la parte superior, la ISO 20000. La certificación de la organización se realiza conforme a la ISO 20000, siendo las personas las que se certifican en ITIL.

### 2.3.2.2. Elementos Principales de ITIL:

La versión 2 de ITIL consta de 7 libros, disponibles en [10], divididos en 2 áreas:

- Gestión de Servicios de TI,
  - Mejores prácticas para la Provisión de Servicio.
  - Mejores prácticas para el Soporte de Servicio.
- Otras guías operativas
  - Gestión de la infraestructura de TI
  - Gestión de la seguridad
  - Perspectiva de negocio
  - Gestión de aplicaciones
  - Gestión de activos de software

Aunque dos de ellos son el núcleo de ITIL V2:

- **Service Support**(Provisión del Servicio). Proporciona una visión táctica del negocio. Engloba los siguientes procesos:
  - Service Level Management (Gestión de Niveles del Servicio): Determina y se responsabiliza de los niveles de servicio que la organización requiere para cumplir con los objetivos del negocio.
  - Financial Management (Gestión Financiera): Gestión financiera de los servicios.
  - Capacity Management (Gestión de la Capacidad): Gestiona que la infraestructura sea la óptima para las necesidades definidas.
  - Continuity Management (Gestión de la Continuidad): Gestiona la continuidad de los servicios.
  - Availability Management (Gestión de la Disponibilidad): Optimiza la disponibilidad de los sistemas.
- **Service Delivery**(Soporte al Servicio). Proporciona una visión operacional. Tiene un componente llamado Centro de Servicios o Service Desk, que debe ser el punto único de entrada y gestión de los servicios. El soporte al servicio, engloba los siguientes procesos:
  - Incident Management (Gestión de Incidentes): Tiene por objetivo restablecer la operativa normal de la organización, tan rápido como sea posible. Un incidente es cualquier circunstancia que provoca que la operativa normal o prestación de servicios no se produzca con la calidad esperada.
  - Problem Management (Gestión de Problemas): Minimiza el impacto de las incidencias y los problemas, buscando las causa a todas las incidencias y previniendo su aparición, tanto de forma proactiva como reactiva.
  - Configuration Management (Gestión de la Configuración): Implementa la CMDB (Configuration Management Data Base) que contiene a los distintos CI (Configuration Items). Los CI son los elementos de la infraestructura que son necesarios para proporcionar el servicio, desde una máquina hasta la definición del responsable del servicio de negocio. Facilita el control del impacto de los CI en la organización, ya que las relaciones entre CI son diseñadas en la CMDB. Gracias a estas relaciones entre estos elementos se puede calcular el impacto en la organización de cada CI ante un posible estado inoperativo del mismo.
  - Change Management (Gestión de Cambios): Gestiona los cambios necesarios en la infraestructura, de forma que se hagan de manera controlada, justificada y autorizada.
  - Release Management (Gestión de Entregas): Gestiona la puesta en producción de los servicios y de los cambios en los CI.

Los libros de la versión 2 de ITIL unían la visión del negocio con la infraestructura, en ITIL V3 se amplía la colección de libros enfocados en el ciclo de vida de los servicios. Esta versión de ITIL consta de 5 libros, también disponibles en el sitio oficial de ITIL [10]. Como se muestra a continuación la v3 amplía los procesos descritos en la v2, no obstante esta versión sigue vigente y se recomienda a las organizaciones empezar por ella si se desea implantar ITIL.

- **Service Strategy**(Estrategia del Servicio): Su objetivo principal es diseñar, desarrollar e implantar la gestión del servicio como un valor de la estrategia corporativa. Se compone de los siguientes procesos:

- Financial Management (Gestión Financiera).
- Demand Management (Gestión de la Demanda).
- Service Portfolio Management (Gestión del Portfolio de Servicios). El portfolio de servicios es el catálogo donde residen todos los servicios de la organización, independientemente de su estado (en explotación o en desarrollo). La parte visible a los clientes del Portfolio es el Catálogo de Servicios, que es el compendio de los Servicios activos (en producción) de la organización.

- **Service Design**(Diseño del Servicio): Su objetivo es diseñar y desarrollar los servicios que cumplan con la estrategia anterior. Se compone de los siguientes procesos:

- Service Catalogue Management (Gestión del Catálogo del Servicio). Define detalles de los servicios existentes y de los procesos de negocio a los que da soporte.
- Service Level Management (Gestión de Niveles de Servicio). Es el proceso responsable de que los niveles pactados del servicio se cumplan.
- Capacity Management (Gestión de la Capacidad). Asegura que la capacidad de los recursos necesarios para proporcionar el servicio con los niveles pactados, sean los adecuados. La capacidad engloba a todos los recursos técnicos, tanto servidores, memoria, etc.
- Availability Management (Gestión de la disponibilidad). Asegura que la disponibilidad de los servicios sean los acordados.
- IT Service Continuity Management (Gestión de la continuidad del Servicio). Asegura la recuperación de los servicios en los niveles pactados.
- Information Security Management (Gestión de la Seguridad de la Información). Se ocupa del cumplimiento de las políticas de seguridad.
- Supplier Management (Gestión de proveedores). Gestiona a los distintos proveedores.

- **Service Transition**(Transición del Servicio): Su objetivo principal es el desarrollo y mejora de los servicios que se desean llevar a producción. Se compone de los siguientes procesos:

- Transition Planning and Support (Planificación y Apoyo de la Transición). Gestiona los recursos involucrados en la puesta en marcha del servicio.
- Change Management (Gestión de Cambios). Gestiona que los cambios estén controlados.
- Service Asset and Configuration Management (Gestión de Activos y Configuración del Servicio). Gestiona y controla los distintos activos necesarios en los servicios.
- Release and Deployment Management (Gestión de Versiones y Despliegue). Construye, libera y despliega los servicios.
- Service Validation and Testing (Validación y Prueba del Servicio). Valida que el servicio cumple con los objetivos establecidos en la definición del servicio.
- Evaluation (Evaluación). Controla si el rendimiento acordado del servicio es el adecuado.
- Knowledge Management (Gestión del Conocimiento). Gestiona que la información sobre el servicio sea conocimiento para todos los involucrados.

- **Service Operation**(Operación del Servicio): Su objetivo es la eficiencia y efectividad de los servicios implantados. Se compone de los siguientes procesos:

- Event Management (Gestión de Eventos). Gestiona los eventos producidos y que son importantes para el servicio.

- Incident Management (Gestión de Incidencias). Gestiona las incidencias producidas para un restablecimiento lo más pronto posible del servicio o un impacto menor.
- Request Fulfillment (Gestión de Peticiones). Gestiona las peticiones de los usuarios.
- Problem Management (Gestión de Problemas). Analiza las causas de las incidencias e intenta que no se vuelvan a producir.
- Access Management (Gestión de Acceso). Gestiona los accesos a los servicios.
- Monitoring and Control (Monitorización y Control). Monitoriza y controla continuamente los servicios.
- IT Operations (Operaciones de TI). Gestiona las actividades necesarias diariamente para la infraestructura del servicio.

• **Continual Service Improvement**(Mejora Continua del Servicio): Enlaza los procesos de Diseño, Transición y operación, con un enfoque de mejora continua. En lugar de un proceso propiamente dicho, son los pasos que hay que realizar para la mejora continua: ¿Qué medir?, ¿Qué se puede medir? Medir, Procesar, Analizar, Informar y ejecutar acciones correctivas.

### 2.3.2.3. Requisitos en ITIL:

Cuando se analiza ITIL para integrar esta propuesta dentro de este trabajo, se observa inicialmente que ITIL se centra en los procesos del servicio. Aun así, sí que incluye los requisitos como mecanismo de explicitar las necesidades de los clientes. Los requisitos son tratados fundamentalmente en el Diseño del Servicio aunque es la Estrategia del Servicio la que genera las necesidades en la Gestión del Portfolio de Servicios.

¿Qué son los objetivos de un proceso sino los requisitos del proceso? ¿Qué es la definición de los servicios que quiero ofrecer sino los requisitos del mismo? ¿Qué son los acuerdos de nivel de servicio sino requisitos?

En el nivel de definición en el que se mueve ITIL, hay dos tipos de requisitos:

- Requisitos de usuario. Es decir, las necesidades que tiene el usuario o los procesos de negocio.
- Requisitos de los procesos. Es decir, qué exigencias tienen los distintos procesos entre sí al interrelacionar en su funcionamiento.

En la figura 9 se observa la relación que tienen los procesos anteriormente definidos.

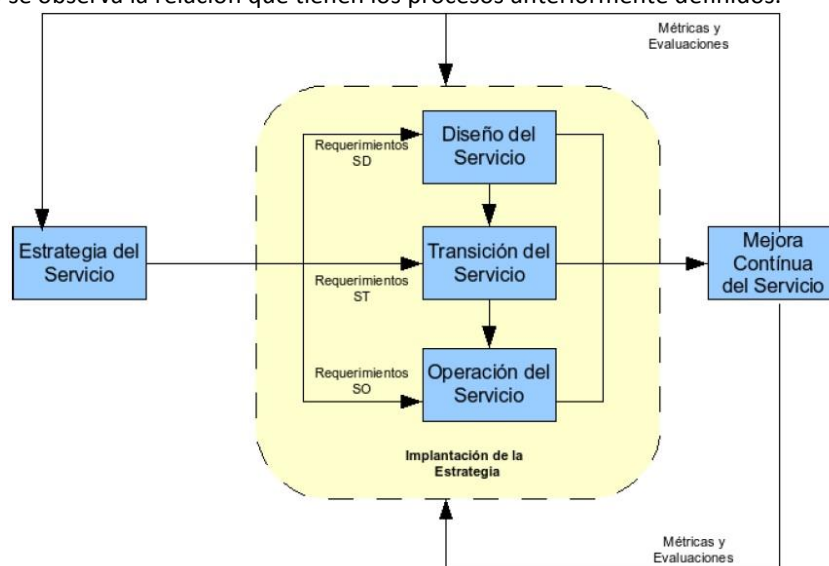


Figura 9: Relaciones entre procesos y requisitos [2]

Conforme un servicio se va definiendo, construyendo e implementando va atravesando los procesos que se definen del ciclo de vida del servicio. Para cada etapa que va atravesando se deben definir los requisitos que se tienen que cumplir. La definición de estas necesidades, es la clave para que se pueda medir y evaluar si se está cumpliendo con lo acordado o necesario. Esta definición de las necesidades es la base para se pueda llevar a cabo un control, y así ante las posibles desviaciones entre lo solicitado y lo realizado se pueda realizar la mejora continua.

ITIL considera los requisitos en los siguientes documentos:

- **Requisitos de Nivel de Servicio** (Service Level Requirement-SLR), que es un documento formal donde se recogen requisitos del Cliente.
- **Criterios de Aceptación del Servicio** (Service Acceptance Criteria-SAC), representan el conjunto de criterios para garantizar que un servicio cumple con la funcionalidad y requisitos de Calidad.
- **Acuerdos de Nivel de Servicio** (Service Level Agreement-SLA), acuerdo entre el proveedor y el cliente.

• **OLA** (Operational Level Agreement-OLA) y **UC** (Underpinning Contract-UC). Son respectivamente los SLA realizados entre partes diferentes dentro de la organización y los UC, son SLA firmados por la organización y un tercero. Estos documentos son fundamentales a la hora de que la provisión del servicio sea traduzca en un proceso formal y no se base en percepción o expectativas subjetivas. La formalización de cualquier elemento dentro del proceso de productos de TI, su elaboración y explotación, son decisivos a la hora de establecer la informática como una ingeniería real. De la misma manera, transformar estos elementos de lenguaje natural a cuestiones computables y medibles dentro del producto es también tarea dentro de la ingeniería.

Las actividades desarrolladas dentro del Service Transition vienen activadas por una **Petición de Cambio (Request for Change-RFC)** cuando el servicio está activo.

En la figura 10 se muestran cómo se van generando los distintos tipos de documentos y por tanto definiendo y ampliando la definición del servicio, en función de la fase en la que se encuentre el producto:

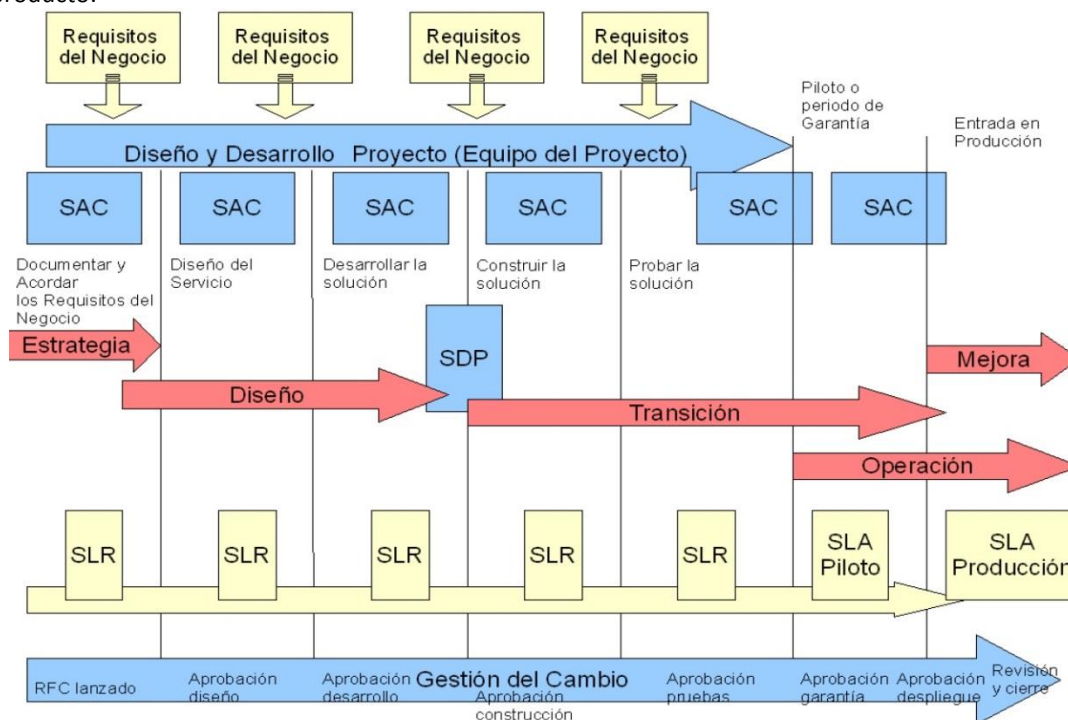


Figura 10: Relaciones entre productos y procesos en ITIL v3 [2].

Una RFC procede por alguna de estas causas:

- Incidencia (Gestión de incidencias)



- Petición. Por cambios en los requisitos, ya sean solicitados por el usuario, mantenimiento adaptativo o perfectivo, cambios en normativa, etc. (Gestión del Servicio, de la Capacidad de la Disponibilidad, etc.)
- Consecuencia de solventar un error conocido (Gestión de Problemas)

Dentro de los datos que se exijan en las RFC, debería aparecer, entre otros:

- Código que identifica a la RFC y al Problema, en caso de proceda de la Gestión de Problemas.
- Descripción del cambio, identificando los CI afectados por el.
- Evaluación del impacto.
- Razón del Cambio.
- Riesgos y consecuencias de no realizar el cambio.
- Prioridad del cambio.
- Recursos necesarios en el cambio
- Aprobación del cambio.
- Plan de vuelta-atrás.

La exigencia que tienen los usuarios de los servicios, le confiere a este proceso una gran importancia, ya que un cambio incontrolado tiene una alta probabilidad de una reducción en la calidad de los servicios, ya sea por un fallo a nivel de infraestructura o por requisitos mal implementados.

Es crucial que dichos cambios sean aprobados, para que se tenga control y respaldo de aquello que se está realizando. Una buena implementación de los procesos de Gestión del Cambio y Gestión de la Configuración reducirán tanto los riesgos, como los costes e impacto que toda evolución de los sistemas conllevan.

Por último, ITIL Application Management, que es el libro de ITIL dónde se aborda el ciclo de desarrollo, proporcionando detalles de cómo ante los cambios del negocio se deben producir definiciones claras de los requisitos y las soluciones aportadas ante las necesidades de los usuarios, clasifica los requisitos de la siguiente forma:

- Functional Requirements (Requisitos Funcionales)
- Non-Functional Requirements (Requisitos No Funcionales)
- Usability Requirements (Requisitos de Usabilidad)
- Change Cases (Casos de Cambio)
- Testing Requirements (Requisitos de Pruebas)
- Requirements management checklist (Lista de gestión de requisitos)
- Organization of the requirements team. (Organización del equipo de requisitos)

### 2.3.3. El proceso de revisión de requisitos:

La revisión de requisitos es una técnica muy poderosa para identificar requisitos ambiguos o incorrectos y realizar un control y validación de todos los requisitos detectados antes de establecer una línea base. Tal y como expone *Karl. Wiegers* en su libro *"Software Requirements"* [8], las revisiones informales, que son las que se realizan normalmente en la cual una persona presumiblemente experta realiza una revisión al documento, son útiles para la educación de otras personas sobre el producto y para la recolección de información no estructurada sobre éste. Existen varias formas de realizar las revisiones informales, entre ellas las más comunes son la revisión del producto de trabajo por un compañero de forma rápida, la revisión del producto por varios compañeros de forma simultánea, o realizar una exposición del producto guiada por el autor del mismo y pedir comentarios y sugerencias a los interlocutores. Las revisiones informales son buenas para la captura de errores graves, inconsistencias y lagunas en las definiciones. Ayudan a detectar estados que no cumplan las características mínimas de calidad de los requisitos.



No obstante, es difícil para un único revisor recoger todos los errores y ambigüedades en una definición de requisitos puesto que tras un tiempo el revisor llega a entender el requisito y no ver ninguna inconsistencia. Otro revisor posteriormente puede volver a leer el requisito y llegar a una interpretación diferente hasta llegar a entender el requisito y volver a no ver más problemas. Si estos dos revisores nunca discuten sobre lo que han visto y revisado en el requisito, las ambigüedades pasaran desapercibidas hasta que aparezca un error en el desarrollo o la utilización del requisito. Una revisión formal de requisitos produce un informe (documento) donde se identificará el material examinado por el revisor y más adelante será supervisado por un equipo de revisores que determinaran si los requisitos revisados en el informe son aceptables. El informe resultante de este ejercicio es un resumen de los defectos encontrados y las cuestiones planteadas por el equipo durante el examen del documento. El tipo de revisión formal más común es el conocido como “inspección”. La inspección del documento de requisitos es una de las técnicas de calidad de software con mayor aceptación en la actualidad. Varias compañías han evitado hasta 10 horas de trabajo (revisando, modificando y subsanando errores) por cada hora dedicada al proceso de revisión mediante inspección. La inspección detallada de grandes conjuntos de requisitos puede ser lenta y tediosa, no obstante, en términos de resultados siempre merece la pena la inversión de esfuerzo y tiempo en el proceso de revisión. Las inspecciones no suelen ser baratas ni rápidas, además de ser una labor tediosa que es evitada por los equipos de desarrollo, sin embargo la alternativa es gastar grandes cantidades de esfuerzo, tiempo y dinero, además de la paciencia de los clientes, resolviendo los problemas que surgirán más adelante.

#### **2.3.3.1. El proceso de inspección:**

El proceso de inspección es un proceso bien definido que consta de varias fases. Es llevado a cabo por un pequeño grupo de participantes que deben examinar detenidamente el producto para identificar defectos o posibles errores. La inspección sirve como una puerta de calidad a través de todas las especificaciones de requisitos que debería ser pasada antes de definir la línea base. A continuación se describe en que consiste el proceso de inspección:

- Participantes:

Hay que asegurarse que se dispone de la gente necesaria realizando una pequeña reunión antes de realizar el procedimiento de inspección. Los participantes en la inspección deben representar 4 perfiles:

1 – Autor del producto: El analista del negocio que ha escrito el documento de requisitos debe proporcionar su perspectiva y conocimientos sobre el mismo. Es recomendable incluir otro analista de requisitos que no haya participado en él puesto que ayudará a identificar errores que al autor se le hayan podido pasar por alto.

2 – Fuentes del requisito: Estos participantes son los representantes del usuario o el propio usuario del cual partieron las necesidades que dieron lugar al requisito. Su presencia asegurará que el requisito cubre todas las necesidades que el usuario tiene en mente.

3 - Desarrolladores del proyecto: Lo ideal sería incluir en este grupo a los desarrolladores, jefes de proyecto, testers y documentadores ya que entre todos pueden detectar gran variedad de tipos de errores. Entre todos los más importantes son los tester y los desarrolladores ya que con sus dos perspectivas se cubren gran parte de las partes de desarrollo.

4 – Responsables de los sistemas: En este grupo se incluye a las personas que pueden detectar errores en las definiciones de los requisitos en las cuales se especifican las interacciones con los sistemas tanto internos como externos.

Se debe tratar de limitar los equipos a grupos de 7 o menos personas. Esto puede implicar que algunos de los perfiles no puedan estar incluidos dentro del equipo de revisión, no obstante los equipos con muchos miembros suelen caer en largas discusiones sobre la mejor resolución de un problema y debates sobre lo que es o no es un error. Acotando los grupos se evitan estas situaciones que dilatan el tiempo del proceso. Otra consideración importante es no incluir en las reuniones del equipo al manager del proyecto, puesto que en estas reuniones se detectan gran cantidad de errores y esto puede llevar al manager a tener una mala impresión sobre los equipos de trabajo.

- Roles:

Todos los participantes en la inspección, incluido el autor, buscará defectos en la especificación. Algunos de los miembros del equipo deberán seguir estos roles específicos durante el proceso de inspección.

- 1- Autor: El autor crea y mantiene el producto que va a ser inspeccionado. El autor del documento de requisitos es, por lo general, un analista que participó en la etapa de educación y conoce las necesidades de las que surgió el requisito. El autor debe anotar y modificar el documento acorde a lo que se revele en la inspección pero no debe participar activamente en el debate.
  - 2- Moderador: El moderador planifica la inspección con el autor, coordina las actividades a realizar y facilita la elaboración de la reunión de inspección. El moderador distribuye el material que debe ser inspeccionado y cualquier documento relevante a los participantes algunos días antes de la ejecución de la revisión para que éstos puedan ir revisándolo y marcando sus anotaciones. El moderador realiza un seguimiento de las modificaciones propuestas con el autor una vez se ha finalizado la reunión para añadirlas correctamente.
  - 3- Lector: Uno de los inspectores debe adjudicarse el rol de lector. El lector leerá en voz alta las especificaciones del requisito y los elementos de modelado examinados al menos una vez. Los otros participantes deben reseñar los potenciales defectos durante la lectura. En ocasiones es recomendable que el lector exponga su punto de vista sobre el requisito para mostrar si difiere del punto de vista de los otros inspectores y así detectar posibles ambigüedades en la descripción. Por lo general el rol de lector suele ser asumido por la persona que ejerce de moderador.
  - 4- Registrador: El registrador utiliza documentos estandarizados para identificar los errores y defectos encontrados durante el proceso de revisión. El registrador debe exponer lo que ha identificado y documentado para aseverar que ha tomado nota de todos los errores expuestos en la reunión. Los inspectores deben ayudar al registrador a identificar y documentar de forma clara los errores detectados.
- Criterios de comienzo: Para saber cuándo es buen momento para realizar una reunión de inspección de un requisito sin hacer perder el tiempo a los participantes y asegurándose de que el documento tiene un nivel suficiente para ser revisado se deben tener en cuenta las siguientes consideraciones:
- 1- El documento se ajusta a una plantilla estándar y no tiene faltas de ortografía ni gramaticales, o problemas con su formato.

- 2- Los números de línea, secciones y referencias están numeradas o codificadas y son fácilmente identificables.
- 3- Todos los temas se marcan como “pendientes” en una herramienta de seguimiento de problemas, o en un documento compartido.
- 4- El moderador no ha encontrado más de tres defectos en 10 minutos tras una revisión del documento.

- Fases: las principales fases del proceso de inspección se muestran en la figura 11:

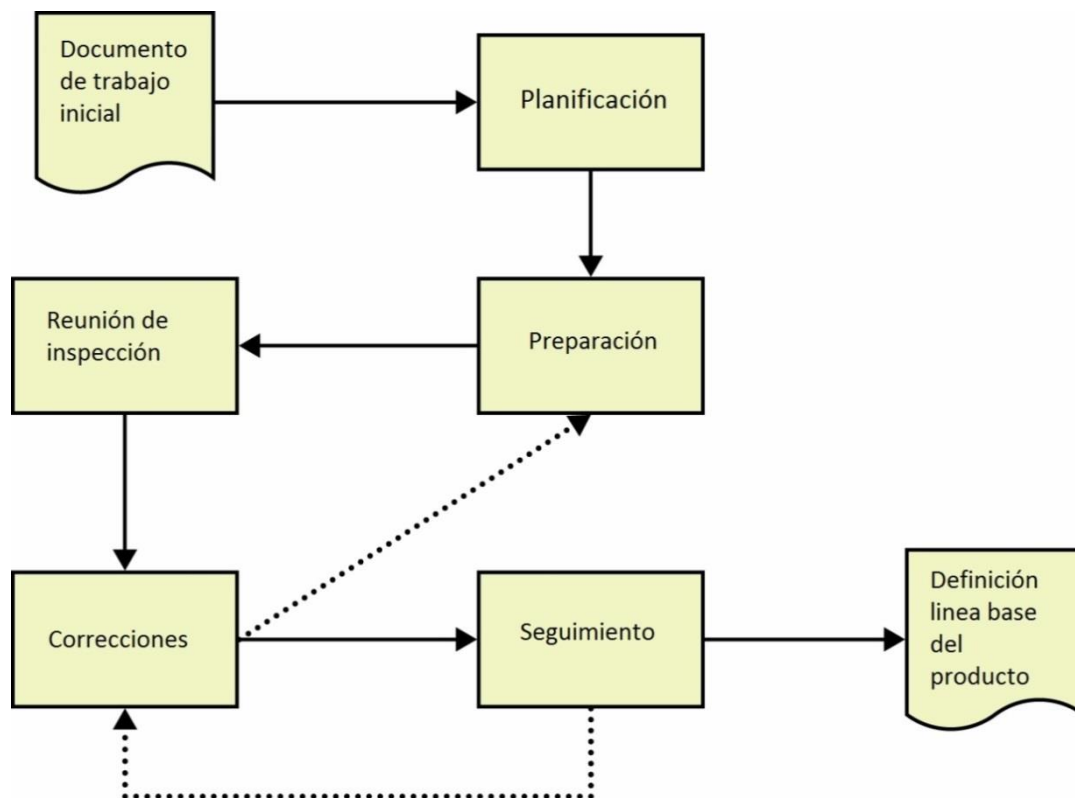


Figura 11: Fases del proceso de inspección (Revisión de Requisitos)

- 1- Fase de Planificación: El autor y el moderador deberán realizar la planificación de la inspección juntos. Ellos determinarán quienes deben participar, que documentos se inspeccionaran, que prioridad recibe cada documento, el tiempo necesario para la reunión, y cuando se debe realizar la inspección. Se debe también definir un número mínimo de páginas revisadas por hora, ya que esto marcará un ritmo de revisión impidiendo que se acabe en divagaciones innecesarias. Hay que tener especial cuidado al marcar esta tasa de páginas revisadas por hora, ya que no dedicar el tiempo suficiente a cada una podría dejar escapar muchos errores, para marcar una buena tasa de revisión hay que tener en cuenta estos factores: el tiempo de revisión previa a la reunión que ha podido dedicar el equipo, la cantidad de información mostrada por página, la complejidad de los requisitos, el impacto de los errores en cada requisito, la criticidad del requisito, el nivel de experiencia de la persona que escribió el requisito.

- 2- Fase de Preparación: Anterior a la reunión, el autor debe suministrar la información necesaria a los inspectores para que tengan conocimientos sobre lo que se va a revisar, y puedan realizar una revisión previa de la documentación identificando posibles errores y problemas más típicos de los requisitos. Es muy importante que cada inspector dedique el tiempo suficiente a la realización de esta fase, ya que aproximadamente el 75% de los errores encontrados en el proceso de inspección son identificados en ese momento. Si cada inspector no ha realizado su preparación es conveniente posponer la reunión de inspección.
- 3- Reunión de Inspección: Durante la reunión el lector realizará una revisión del requisito con sus propias palabras, los inspectores deberán exponer los errores y defectos detectados en cada requisito y el autor deberá apunrarlos para realizar sus correcciones. Es conveniente no realizar reuniones de más de 2 horas ya que el cansancio reduce la capacidad de detectar los errores y los requisitos. Si no se dispone de tiempo suficiente, es recomendable realizar reuniones adicionales que alargar el tiempo. Una vez revisado un documento el equipo de revisión decidirá si se acepta el documento con las correcciones identificadas, o si será necesaria una reelaboración del mismo dado su grado de incorrección.
- 4- Fase de Reworking: Durante esta fase el autor debe pasar algún tiempo volviendo a trabajar en el requisito, corrigiendo errores y eliminando las ambigüedades o los defectos que pudiera encontrar.
- 5- Fase de Seguimiento: En la fase de seguimiento el moderador debe asegurarse de que todas las cuestiones planificadas han sido abordadas en la reunión y todos los documentos han cumplido las condiciones de cierre. Si una de las modificaciones es incompleta o no se ha realizado una revisión correcta del documento se dará lugar a otra reunión.  
Las condiciones para cerrar un documento son: todas las cuestiones planteadas durante la inspección han sido resueltas, cualquiera de los cambios realizados en los requisitos se han llevado a cabo de forma correcta y todos los temas pendientes se han cerrado y documentado.

#### **2.3.3.2. Listado de control de defectos:**

Para ayudar a los revisores a encontrar los errores más típicos en los productos se realizan listas de los tipos de defectos más comunes para que los revisores puedan identificar rápidamente qué deben identificar en el documento. Transcurrido un tiempo los revisores interiorizaran los conceptos no haciendo falta utilizar este listado, pero para revisores noveles es una herramienta que puede ser muy útil. En la figura 12 se muestra una para la lista de control de defectos.

### Completeness

- ☐ Do the requirements address all known customer or system needs?
- ☐ Is any needed information missing? If so, is it identified as TBD?
- ☐ Have algorithms intrinsic to the functional requirements been defined?
- ☐ Are all external hardware, software, and communication interfaces defined?
- ☐ Is the expected behavior documented for all anticipated error conditions?
- ☐ Do the requirements provide an adequate basis for design and test?
- ☐ Is the implementation priority of each requirement included?
- ☐ Is each requirement in scope for the project, release, or iteration?

### Correctness

- ☐ Do any requirements conflict with or duplicate other requirements?
- ☐ Is each requirement written in clear, concise, unambiguous, grammatically correct language?
- ☐ Is each requirement verifiable by testing, demonstration, review, or analysis?
- ☐ Are any specified error messages clear and meaningful?
- ☐ Are all requirements actually requirements, not solutions or constraints?
- ☐ Are the requirements technically feasible and implementable within known constraints?

### Quality Attributes

- ☐ Are all usability, performance, security, and safety objectives properly specified?
- ☐ Are other quality attributes documented and quantified, with the acceptable trade-offs specified?
- ☐ Are the time-critical functions identified and timing criteria specified for them?
- ☐ Have internationalization and localization issues been adequately addressed?
- ☐ Are all of the quality requirements measurable?

### Organization and Traceability

- ☐ Are the requirements organized in a logical and accessible way?
- ☐ Are all cross-references to other requirements and documents correct?
- ☐ Are all requirements written at a consistent and appropriate level of detail?
- ☐ Is each requirement uniquely and correctly labeled?
- ☐ Is each functional requirement traced back to its origin (e.g., system requirement, business rule)?

### Other Issues

- ☐ Are any use cases or process flows missing?
- ☐ Are any alternative flows, exceptions, or other information missing from use cases?
- ☐ Are all of the business rules identified?
- ☐ Are there any missing visual models that would provide clarity or completeness?
- ☐ Are all necessary report specifications present and complete?

Figura 12: Ejemplo de lista de control de defectos [8].

Cada organización es libre de realizar sus listas de control de defectos acorde con sus necesidades, no obstante cuanto más elaboradas y extensas son estas listas mayor impacto tendrán en la calidad de los requisitos revisados.

#### 2.3.3.3. Buenas prácticas durante la revisión de requisitos:

- Planificar las revisiones no implica leer un documento de principio a fin, es necesario centrar a los revisores en puntos específicos de cada documento.
- Comenzar a revisar los documentos de especificaciones del requisito antes de que éste esté completo. No es necesario que la especificación de un requisito este realizada al 100% para empezar a revisar algunas de sus secciones.
- Asignar el tiempo suficiente es crítico para que los revisores hagan una inspección adecuada de cada requisito. Por términos generales los proyectos se desarrollan con tiempos muy ajustados, pero no dedicar tiempo al proceso de revisión es caer en un error.
- Proporcionar un contexto a los revisores facilitará mucho su tarea si éstos no trabajan dentro del proyecto y no conocen sus implicaciones.

- Determinar el alcance de las revisiones durante la planificación determinando que partes se deben revisar concentrará mucho el trabajo de los revisores evitando pérdidas de tiempo y aumentará su efectividad.
- Evitar que la misma persona revise el documento más de tres veces. Si una persona no ha sido capaz de ver un defecto o error en ese punto, es muy probable que no lo haga en revisiones posteriores. En este caso, es mejor que otra persona diferente revise el documento.
- Dar prioridad a determinadas áreas dentro de un mismo documento ayudará a que los errores más críticos estén corregidos en primera instancia dejando las áreas secundarias para más adelante si no pueden ser abordadas.

### **2.3.4. Marcos y Normativas:**

Actualmente no existe un estándar de facto que proporcione una orientación o procedimiento exacto de cómo gestionar requisitos, ni que se encuentren libres de defectos ni de críticas por otros autores, sin embargo existen normativas dirigidas a formalizar el proceso de ingeniería de requisitos en determinados aspectos diseminados a lo largo del ciclo de vida y también existen determinados procesos de ciertas normas o estándares que utilizan el proceso de requisitos como una sección dentro del proceso de creación de un proyecto software.

A continuación se realiza una exposición de una selección de estándares disponibles realizando una revisión a sus contenidos.

#### **2.3.4.1. IEEE 830:1998: Recommended Practice for Software Requirements Specifications:**

La normativa IEEE 830:1998 reúne un conjunto de acciones recomendadas en la elaboración de una Especificación de Requisitos Software. No obstante esta normativa es actualmente sustituida por la norma IEEE 29148-2011 (expuesta más adelante en este documento) por ser ésta más completa y actual, sin ser una normativa específica sobre el mismo tema, puesto que la IEEE 29148-2011 hace referencia al ciclo de vida de los requisitos.

##### **1. Introducción**

En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software (ERS). Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

##### **1.1. Propósito**

En esta subsección se definirá el propósito del documento ERS y se especificará a quien va dirigido el documento.

##### **1.2. Ámbito del Sistema**

En esta subsección:

Se podrá dar un nombre al futuro sistema (p.ej. Mi Sistema)

Se explicará lo que el sistema hará y lo que no hará.

Se describirán los beneficios, objetivos y metas que se espera alcanzar con el futuro sistema.

Se referenciarán todos aquellos documentos de nivel superior (p.e. en Ingeniería de Sistemas, que incluyen Hardware y Software, deberá mantenerse la consistencia con el documento de especificación de requisitos globales del sistema, si existe).

### 1.3. Definiciones, Acrónimos y Abreviaturas

En esta subsección se definirán todos los términos, acrónimos y abreviaturas utilizadas en la ERS.

### 1.4. Referencias

En esta subsección se mostrara una lista completa de todos los documentos referenciados en la ERS.

### 1.5. Visión General del Documento

Esta subsección describe brevemente los contenidos y la organización del resto de la ERS.

## 2. Descripción General

En esta sección se describen todos aquellos factores que afectan al producto

Y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender.

Normalmente, esta sección consta de las siguientes subsecciones: Perspectiva del producto, funciones del producto, características de los usuarios, restricciones, factores que se asumen y futuros requisitos.

### 2.1. Perspectiva del Producto

Esta subsección debe relacionar el futuro sistema (producto software) con otros productos. Si el producto es totalmente independiente de otros productos, también debe especificarse aquí. Si la ERS define un producto que es parte de un sistema mayor, esta subsección relacionara los requisitos del sistema mayor con la funcionalidad del producto descrito en la ERS, y se identificaran las interfaces entre el producto mayor y el producto aquí descrito.

Se recomienda utilizar diagramas de bloques.

### 2.2. Funciones del Producto

En esta subsección de la ERS se mostrara un resumen, a grandes rasgos de las funciones del futuro sistema. Por ejemplo, en una ERS para un programa de contabilidad, esta subsección mostrara que el sistema soportara el mantenimiento de cuentas, mostrara el estado de las cuentas y facilitara la facturación, sin mencionar el enorme detalle que cada una de estas funciones requiere.

Las funciones deberán mostrarse de forma organizada, y pueden utilizarse gráficos, siempre y cuando dichos gráficos reflejen las relaciones entre funciones y no el diseño del sistema.

### 2.3. Características de los Usuarios

Esta subsección describirá las características generales de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.

### 2.4. Restricciones

Esta subsección describirá aquellas limitaciones que se imponen sobre los desarrolladores del producto:

- Políticas de la empresa
- Limitaciones del hardware
- Interfaces con otras aplicaciones
- Operaciones paralelas
- Funciones de auditoria
- Funciones de control
- Lenguaje(s) de programación
- Protocolos de comunicación
- Requisitos de habilidad
- Criticidad de la aplicación
- Consideraciones acerca de la seguridad

### 2.5. Suposiciones y Dependencias

Esta subsección de la ERS describirá aquellos factores que, si cambian pueden afectar a los requisitos. Por ejemplo, los requisitos pueden presuponer una cierta organización de ciertas unidades de la empresa, o pueden presuponer que el sistema correrá sobre cierto sistema operativo. Si cambian dichos



detalles en la organización de la empresa, o si cambian ciertos detalles técnicos, como el sistema operativo, puede ser necesario revisar y cambiar los requisitos.

### **2.6. Requisitos Futuros**

Esta subsección esbozará futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.

### **3. Requisitos Específicos**

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses. Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.

Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.

Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:

- Corrección: La ERS es correcta si y solo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.

- No ambiguos: Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales.

En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.

- Completo: Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válido como no válido.

- Consistentes: Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.

- Clasificados: Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esencial, condicional u opcional) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo para no emplear excesivos recursos en implementar requisitos no esenciales.

- Verificables: La ERS es verificable si y solo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc. Introducen ambigüedad en los requisitos. Requisitos como "en caso de accidente la nube tóxica no se extenderá más allá de 25Km" no es verificable por el alto costo que conlleva.

- Modificables: La ERS es modificable si y solo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos (por ejemplo RequisitePro o Doors) facilitan enormemente esta tarea.

- Trazables: La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica que componentes del sistema son los que realizan el requisito R.

#### **3.1. Interfaces Externas**

Se describirán los requisitos que afecten a la interfaz de usuario, interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones.

#### **3.2. Funciones**

Esta subsección (quizá la más larga del documento) deberá especificar todas aquellas acciones (funciones) que deberá llevar a cabo el software. Normalmente (aunque no siempre), son aquellas



acciones expresables como “el sistema deberá. . .”. Si se considera necesario, podrán utilizarse notaciones gráficas y tablas, pero siempre supeditadas al lenguaje natural, y no al revés.

Es importante tener en cuenta que, en 1983, el Estándar de IEEE 830 establecerá que las funciones deberían expresarse como una jerarquía funcional (en paralelo con los DFDs propuestos por el análisis estructurado). Pero el Estándar de IEEE 830, en sus últimas versiones, ya permite organizar esta subsección de múltiples formas, y sugiere, entre otras, las siguientes:

- Por tipos de usuario: Distintos usuarios poseen distintos requisitos. Para cada clase de usuario que exista en la organización, se especificarán los requisitos funcionales que le afecten o tengan mayor relación con sus tareas.
- Por objetos: Los objetos son entidades del mundo real que serán reflejadas en el sistema. Para cada objeto, se detallarán sus atributos y sus funciones. Los objetos pueden agruparse en clases. Esta organización de la ERS no quiere decir que el diseño del sistema siga el paradigma de Orientación a Objetos.
- Por objetivos: Un objetivo es un servicio que se desea que ofrezca el sistema y que requiere una determinada entrada para obtener su resultado. Para cada objetivo o subobjetivo que se persiga con el sistema se detallarán las funciones que permitan llevarlo a cabo.
- Por estímulos: Se especificarán los posibles estímulos que recibe el sistema y las funciones relacionadas con dicho estímulo.
- Por jerarquía funcional: Si ninguna de las anteriores alternativas resulta de ayuda, la funcionalidad del sistema se especificará como una jerarquía de funciones que comparten entradas, salidas o datos internos. Se detallarán las funciones (entrada, proceso, salida) y las subfunciones del sistema. Esto no implica que el diseño del sistema deba realizarse según el paradigma de Diseño Estructurado. Para organizar esta subsección de la ERS se elegirá alguna de las anteriores alternativas, o incluso alguna otra que se considere más conveniente.

Deberá, eso sí, justificarse el porqué de tal elección.

### 3.3. Requisitos de Rendimiento

Se detallarán los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc.

También, si es necesario, se especificarán los requisitos de datos, es decir aquellos requisitos que afecten a la información que se guardara en la base de datos. Por ejemplo, la frecuencia de uso, las capacidades de acceso y la cantidad de registros que se espera almacenar (decenas, cientos, miles o millones).

### 3.4. Restricciones de Diseño

Todo aquello que restrinja las decisiones relativas al diseño de la aplicación: Restricciones de otros estándares, limitaciones del hardware, etc.

### 3.5. Atributos del Sistema

Se detallarán los atributos de calidad (las “habilidades”) del sistema: Fiabilidad, mantenibilidad, portabilidad, y, muy importante, la seguridad. Deberá especificarse que tipos de usuario están autorizados, o no, a realizar ciertas tareas y como se implementarán los mecanismos de seguridad (por ejemplo, por medio de un login y una password).

### 3.6. Otros Requisitos

Cualquier otro requisito que no encaje en otra sección.

## 4. Apéndices

Pueden contener todo tipo de información relevante para la ERS pero que, propiamente, no forme parte de la ERS. Por ejemplo:

1. Formatos de entrada/salida de datos, por pantalla o en listados.
2. Resultados de análisis de costes.
3. Restricciones acerca del lenguaje de programación.

### 2.3.4.2. ISO/IEC/IEEE 29148:2011: Systems and software engineering – Life cycle processes – Requirements engineering:

La norma ISO/IEC/IEEE 29148 es un compendio de diferentes normativas tanto de requisitos como de procesos de desarrollo de productos software en la cual se hace enfoque sobre el proceso de ingeniería de requisitos. Las normativas de las que hereda la ISO/IEC/IEEE 29148 son:

- ISO/IEC/IEEE 12207:2008 -*Systems and software engineering – Software life cycle processes.*
- ISO/IEC/IEEE 15288:2008 – *Systems and software engineering – Software life cycle processes.*
- ISO/IEC/IEEE 15289:2011 – *Systems and software engineering – Content of life-cycle information products (documentation).*
- ISO/IEC TR 19759 – *Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOK)*
- IEEE Std 830 (anteriormente comentada) – *IEEE Recommended Practice for Software Requirements Specifications.*
- IEEE Std 1233 – *IEEE Guide for Developing System Requirements Specifications*
- ISO/IEC TR 24748-1 – *Systems and software engineering – Life cycle management – Part 1: Guide for life cycle management.*
- ISO/IEC/IEEE 24765 – *Systems and software engineering – Vocabulary.*

Su objetivo es especificar los procesos necesarios que deben ser implementados para la ingeniería de requisitos para sistemas y productos software (incluidos los servicios) en todo el ciclo de vida, dar pautas para la aplicación de los requisitos y procesos de requisitos relacionados con las prácticas indicadas en las normas ISO/IEC 12207 e ISO/IEC 25288, especificar los elementos de información de los requisitos que deben ser producidos a través de la aplicación del proceso de requisitos, especificar el contenido requerido de los elementos de información adicionales y dar pautas para el formato de los elementos de información relacionados a los requisitos.

Dentro del conjunto de estándares y normativas relativas a Ingeniería de Requisitos se considera a la ISO/IEC/IEEE 29148 la más completa en la actualidad y por tanto es la más utilizada por las organizaciones desarrolladoras.

Las principales secciones de esta normativa son:

#### **1. Conceptos:**

##### **1.1. Fundamentos de requisitos:**

En esta sección se describen los conceptos fundamentales utilizados en los requisitos tales como que son los Stakeholders, cómo se transforman las necesidades en requisitos, cómo se debe construir un requisito, características individuales de cada requisito, características grupales de los requisitos, criterios de lenguaje dentro de los requisitos y como asignar atributos a los requisitos.

##### **1.2. Consideraciones prácticas:**

En esta sección se describen dos vías o métodos para la obtención de los requisitos: la aplicación iterativa de procesos y la aplicación recursiva de procesos.

### **1.3. Elementos de información de requisitos:**

En esta sección se describe como relacionar los diferentes elementos de información con los requisitos.

## **2. Procesos:**

### **2.1. Procesos de requisitos:**

En esta parte del documento se exponen unas pequeñas guías de cómo se deben crear los requisitos (en sus fases de definición y análisis) según la normativa ISO/IEC 15288:2008.

### **2.2. Proceso de definición de requisitos de usuario ("Stakeholders"):**

En esta sección se describe cual es el proceso de obtención de los requisitos de usuarios u otros posible implicados en el proyecto ("stakeholders"). Se explica las consideraciones que se deben tener en cuenta tanto en el proceso de captura de los requisitos como en su posterior definición y documentación respetando las indicaciones de la norma ISO/IEC 15288:2008.

### **2.3. Proceso de análisis de requisitos:**

Sección en la cual se describen las tareas a llevar a cabo en el proceso de análisis de los requisitos para dejarlos correctamente documentados y preparados. Entre estas tareas encontramos como realizar un refinamiento de las definiciones de los requisitos, analizar y mantener la integridad del sistema de requisitos, definir procesos de validación de los requisitos, prototipados, procesos de revisión, modelado y simulación, detección de problemas, definición de la trazabilidad y la jerarquía entre los requisitos ...etc.

### **2.4. Actividades de ingeniería de requisitos en otros procesos técnicos:**

En esta sección se tienen en cuenta los procesos adicionales que serán necesarios para la captura y definición de los requisitos, entre estos procesos destacan los siguientes:

- Requisitos en el diseño de la arquitectura: Se describe los procesos de diseño, análisis y evaluación de la arquitectura que se utilizará en el sistema.
- Verificación de requisitos: Se describe como realizar un plan de verificación basado en el sistema de requisitos ya identificados y las diferentes vías de verificación (inspección, análisis, tests, demostraciones...etc).
- Validación de requisitos: Se describe como debe realizarse el proceso de validación de los requisitos, quien debe hacerlo y como consolidar las validaciones.

### **2.5. Gestión de requisitos:**

En esta parte del documento se exponen las diferentes tareas a realizar en la gestión de requisitos. Tras realizar una exposición sobre lo que significa gestión de requisitos se resaltan las siguientes tareas:

- Gestión de cambios: Se describen que acciones se deben realizar y tener en cuenta cuando surge uno o varios cambios en los requisitos y como pueden afectar al resto de requisitos.
- Gestión de configuración: En esta sección se describe como definir y gestionar un plan de configuración (mantener la información sobre las configuraciones con un nivel adecuado de integridad y seguridad).

- Gestión de información: En esta sección se describe cómo se deben gestionar los diferentes elementos que aportan información sobre o acerca de los requisitos.
- Medición de requisitos: El propósito de esta sección es recoger, analizar y reportar los datos relativos a los componentes desarrollados e implementados dentro del proyecto, para apoyar una gestión eficaz y demostrar de manera objetiva la calidad de los componentes desarrollados.

### 3. Elementos de Información:

En el proceso de la elaboración de la documentación de los requisitos se producen principalmente estos tres tipos de documento:

- 1 – Documento de especificación de requisitos de usuarios o partes interesadas (“Stakeholders”) por sus siglas en inglés “StRS” (Stakeholder Requirements Specification).
- 2 – Documento de especificación de requisitos del sistema, por sus siglas en “inglés” SyRS (System Requirements Specification).
- 3 – Documento de especificación de requisitos de software, o por sus siglas en inglés “SRS” (Software Requirements Specification).

#### 3.1. Documento de especificación de requisitos de Stakeholders:

En la sección se muestra debe redactarse la documentación para describir los requisitos que surgen por las necesidades de los usuarios o de las partes implicadas en el proceso a desarrollar mostrando un ejemplo de la estructura que debe seguir el documento y el contenido que debe reflejar cada sección. Habitualmente estos documentos reflejan los requisitos de usuario, los requisitos de negocio y los requisitos organizacionales acorde a la normativa ISO/IEC/IEEE 15289 mostrando un ejemplo de la estructura que debe seguir el documento y el contenido que debe reflejar cada sección. En la figura 13 se muestra un ejemplo de documento de especificación de requisitos de stakeholders.

<b>1. Introduction</b>
1.1 Business purpose
1.2 Business scope
1.3 Business overview
1.4 Definitions
1.5 Stakeholders
<b>2. References</b>
<b>3. Business management requirements</b>
3.1 Business environment
3.2 Goal and objective
3.3 Business model
3.4 Information environment
<b>4. Business operational requirements</b>
4.1 Business processes
4.2 Business operational policies and rules
4.3 Business operational constraints
4.4 Business operational modes
4.5 Business operational quality
4.6 Business structure
<b>5. User requirements</b>
<b>6. Concept of proposed system</b>
6.1 Operational concept
6.2 Operational scenario
<b>7 Project Constraints</b>
<b>8. Appendix</b>
8.1 Acronyms and abbreviations

Figura 13: Ejemplo de documento StRS [13].

### 3.2. Documento de especificación de requisitos del sistema:

En la sección se muestra cómo redactarse la documentación para describir los requisitos que surgen por las necesidades técnicas del proyecto, las especificaciones de iteración entre usuario-sistema y las necesidades de usabilidad mostrando un ejemplo de la estructura que debe seguir el documento y el contenido que debe reflejar cada sección. En la figura 14 se muestra una plantilla para un documento de especificación de requisitos del sistema.

<b>1. Introduction</b>
1.1 System purpose
1.2 System scope
1.3 System overview
1.3.1 System context
1.3.2 System functions
1.3.3 User characteristics
1.4 Definitions
<b>2. References</b>
<b>3. System requirements</b>
3.1 Functional requirements
3.2 Usability requirements
3.3 Performance requirements
3.4 System interface
3.5 System operations
3.6 System modes and states
3.7 Physical characteristics
3.8 Environmental conditions
3.9 System security
3.10 Information management
3.11 Policies and regulations
3.12 System life cycle sustainment
3.13 Packaging, handling, shipping and transportation
<b>4. Verification</b>
(parallel to subsections in Section 3)
<b>5. Appendices</b>
Assumptions and dependencies

Figura 14: Ejemplo de documento SyRS [13].

### 3.4. Documento de especificación de requisitos de software:

En la sección se muestra cómo debe redactarse la documentación para describir los requisitos de software mostrando un ejemplo de la estructura que debe seguir el documento y el contenido que debe reflejar cada sección. En la figura 15 se muestra una plantilla para un documento de especificación de requisitos del software.

<b>1. Introduction</b>
1.1 Purpose
1.2 Scope
1.3 Product overview
1.3.1 Product perspective
1.3.2 Product functions
1.3.3 User characteristics
1.3.4 Limitations
1.4 Definitions
<b>2. References</b>
<b>3. Specific requirements</b>
3.1 External interfaces
3.2 Functions
3.3 Usability Requirements
3.4 Performance requirements
3.5 Logical database requirements
3.6 Design constraints
3.7 Software system attributes
3.8 Supporting information
<b>4. Verification</b>
(parallel to subsections in Section 3)
<b>5. Appendices</b>
5.1 Assumptions and dependencies
5.2 Acronyms and abbreviations

Figura 15: Ejemplo de documento SRS [13].

#### **2.3.4.3. ISO/IEC TR 24766 – Information technology – Systems and software engineering – Guide for requirements engineering tool capabilities:**

Aunque la norma 24766 no es como tal un estándar internacional, se trata de un “Technical Report” de tipo 2, es decir, un documento que se encuentra en proceso de convertirse en un estándar internacional pero por algún motivo aún no ha conseguido esa categoría. Aunque se apoya en diversas normativas en las cuales se describen procesos y actividades sobre requisitos principalmente *ISO/IEC 12207:2008 – Systems and software engineering-Software life cycle processes* y *ISO/IEC 15288:2008 Systems and software engineering-Software life cycle processes*, la norma 24766 se enfoca en las capacidades y habilidades que deben contener las herramientas de ingeniería de requisitos para dar soporte y calidad durante todo el ciclo de vida de éstos. La norma ISO/IEC TR 23766 está disponible en [28].

Las secciones que contiene esta norma son:

##### **1. Capacidades de las herramientas de requisitos:**

En esta sección se describen por etapas de desarrollo las diferentes capacidades o habilidades que deben facilitar las herramientas de requisitos.

###### **1.1. Educción de requisitos:**

En la sección de educación de requisitos se exponen las habilidades o capacidades necesarias en las herramientas para realizar las diferentes acciones habituales en esta etapa del ciclo de vida de los requisitos.

Dentro de esta etapa se divide en diversos grupos de acciones:

- Captura de requisitos: Acciones a realizar durante la obtención de los requisitos a partir de sus orígenes (stakeholders, arquitectura, normativas...etc.).
- Captura “tal-cual” de elementos del sistema: Acciones a realizar en la captura de elementos funcionales o de calidad de los requisitos que no parten de una fuente específica sino que son necesarios de por sí.
- Trazabilidad de requisitos y stakeholders: Acciones que la herramienta debe facilitar para las relaciones y trazabilidad entre los requisitos y sus “Stakeholders”.
- Escenarios orientados a objetivos y modelados de alto nivel: Utilidades que las herramientas deben contener para facilitar la creación de escenarios orientados a objetivos y realizar modelados de alto nivel.
- Plantillas: Utilidades que debe facilitar la herramienta para la disposición de plantillas para la captura de requisitos.
- Prototipos: Utilidades que debe facilitar la herramienta para la elaboración de prototipos.
- Importación/Exportación de fuentes: Facilidades que debe ofrecer la herramienta para la importación o exportación de elementos a otras herramientas o sistemas.
- Documentación de la educación: Utilidades que debe disponer la herramienta para la correcta gestión y almacenamiento de los documentos elaborados en esta etapa.

### 1.2. Análisis de Requisitos:

En esta sección se describen las capacidades que debe facilitar la herramienta en la etapa de análisis de los requisitos.

Encontramos las siguientes subsecciones dentro del documento:

- Análisis de los requisitos funcionales: Capacidades que la herramienta debe ofrecer para realizar el análisis de los requisitos funcionales.
- Análisis de los requisitos de calidad: Utilidades que la herramienta debe disponer para la realización del análisis de los requisitos de calidad.
- Análisis de viabilidad: La herramienta debe contener utilidades que faciliten un rápido vistazo de la viabilidad en la elaboración de un requisito.
- Modelado: Facilidades para el modelado de los requisitos, principalmente diagramas UML, relacionales, de flujo...etc.
- Prototipos: Utilidades para ofrecer prototipos de interfaces o algoritmos.
- Análisis de atributos: Utilidades para poder realizar análisis sobre los atributos asignados a los requisitos.
- Refinamiento de requisitos: Utilidades de análisis semántico y relacional entre los requisitos para realizar correcciones y mejorar su calidad.
- Análisis de riesgos: La herramienta debe permitir realizar un análisis de los riesgos asociados a cada requisito.
- Métodos de decisión: La herramienta debe disponer de métodos para reflejar los criterios de los diversos stakeholders del proyecto sobre la aceptación o no de un requisito.



- Artefactos de análisis de los requisitos: Utilidades para la exportación en diversos formatos de la documentación de los requisitos para generarlos como elementos externos a la herramienta.

### **1.3. Especificación de requisitos:**

La sección de especificación de requisitos provee de las capacidades y funciones que un sistema debe respetar.

Encontramos las siguientes subsecciones dentro del documento:

- Especificaciones de desarrollo: Utilidades para mostrar en la especificación de requisitos las descripciones de todos los estados posibles dentro de un requisito y bajo qué condiciones se dan estos estados.
- Análisis de trazabilidad: Durante el análisis de requisitos los cambios deben ser documentados, la trazabilidad establece relaciones entre requisitos, fuentes y el estatus general del proyecto. La herramienta debe dar soporte para realizar el análisis de esta trazabilidad.
- Documentación de la especificación de requisitos: Las salidas de todo el sistema de requisitos debe estar almacenada y permitir su exportación en varios formatos.

### **1.4. Verificación y validación de requisitos:**

En esta sección se muestran las facilidades que la herramienta debe ofrecer al usuario para realizar la verificación y validación de los requisitos de una forma cíclica e iterativa a lo largo de todo el ciclo de vida del proyecto e involucrando a todos los implicados necesarios del proyecto en el proceso de verificación y validación de un requisito.

Encontramos las siguientes subsecciones dentro del documento:

- Revisión e Inspección: Sección en la que se indican capacidades que las herramientas deben suministrar para controlar y evitar ambigüedades, mal uso del lenguaje, manejar comentarios y reportes, suministrar y almacenar un histórico de revisiones y resultados de inspecciones.
- Verificación: La herramienta debe dar soporte para el proceso de verificación proveyendo de casos de verificación a los requisitos, generación de reportes de excepciones y suministrando planes de procedimientos de verificación.
- Validación: Las herramientas deben facilitar el proceso de verificación dando soporte para la realización de casos de validación, generación de reportes de excepciones de requisitos que no pases un determinado plan de validación, generación de planes de validación y seguimiento de la trazabilidad del proceso de validación de un requisito.
- Definición de criterios de aceptación: La herramienta debe permitir introducir y gestionar una serie de criterios de aceptación dentro de los planes de validación y verificación para poder considerar validado a un requisito.
- Línea Base de requisitos: Una vez que todos los requisitos han sido validados, la herramienta debe establecer una línea base de esos requisitos y realizar un control de cambios y versiones de cada uno.

### 1.5. Gestión de Requisitos:

La gestión de los requisitos es una actividad que debe realizarse durante todo el ciclo de vida del requisito. Las herramientas deben permitir monitorizar todos los cambios realizados y aportar facilidades para llevar a cabo el mantenimiento de cada requisito hasta que este ha sido completado.

Dentro de esta sección del requisito se encuentran los apartados:

- Identificación de los elementos de configuración de los requisitos: Después de que la línea base ha sido establecida los cambios tienen que ser fácilmente visualizables gestionando el número de versión, fecha, usuario que cambio los requisitos y permitiendo navegar entre todas las versiones del requisito.
- Gestión de los cambios de requisitos: La herramienta debe facilitar un canal de comunicación entre todos los implicados en un requisito para transmitirles cada cambio que este ha sufrido.
- Gestión de la trazabilidad: La herramienta debe permitir la asociación de la documentación del cambio dentro del requisito.
- Gestión de riesgos: La herramienta debe permitir la identificación y gestión de riesgos que surjan a lo largo del ciclo de vida del requisito, permitiendo su asociación a un requisito que lo mitigue.
- Gestión del proyecto: La herramienta debe permitir realizar una gestión global del proyecto facilitando informes generales sobre el estado de todos los requisitos.

### 1.6. Otras capacidades de las herramientas:

En esta sección se listan las características que deben contener las herramientas que no tienen cabida en el resto de secciones.

Entre ellas están:

- Acceso colaborativo: La herramienta debe definir un sistema de roles de sus usuarios permitiendo que cada uno realice sus funciones específicas dentro de cada requisito en colaboración con el resto de implicados en el requisito.
- Información administrativa: La herramienta debe facilitar información adicional del proyecto tales como su magnitud, número de usuarios, numero de desarrolladores, numero de analistas, hardware requerido, información de la base de datos..etc.
- GUI: La herramienta debe dar soporte para la realización de interfaces gráficas.

## 2. Características de Calidad en los requisitos:

En este apartado del documento se listan una serie de características de calidad que las herramientas deben utilizar para detectar y corregir errores en los requisitos de forma temprana.

### 2.1. Atributos de calidad de los requisitos:

Se muestra un conjunto de atributos de calidad que deben contener los requisitos y se explica en que consiste cada uno. Estos atributos son: singularidad, corrección, concisión, no ambigüedad,

trazabilidad, obligatoriedad, prioridad, verificabilidad, abstracción, factibilidad, estabilidad, suficiencia, necesidad, consistencia, uniformidad, modularidad, alcance, unicidad, complejidad, entendimiento, abordabilidad.

## **2.2. Habilidades de las herramientas para las características de los requisitos:**

En esta sección se abordan las utilidades que deben disponer las herramientas para ayudar a cumplir los atributos de calidad mencionados en la sección anterior.

### **2.3.4.4. ISO/IEC 15288 – Systems and software engineering – Systems life cycle processes:**

La norma 15288 constituye un estándar internacional de guías de aplicación en los procesos y actividades a realizar en cada etapa del ciclo de vida del desarrollo de un sistema. Su principal objetivo es mejorar la comunicación entre peticionarios, suministradores y cualquier implicado en el desarrollo del proyecto en cada uno de sus procesos del ciclo de vida. Abarca desde la etapa de concepción del producto hasta su retirada. Los procesos que compete se describen en un bloque llamado “Ciclo de vida de los procesos de sistemas”.

#### **1. Ciclo de vida de los procesos de sistemas:**

La norma dice que estos procesos son iterativos, es decir, no necesariamente deben seguir el orden en el que aparecen y se pueden aplicar tantas veces sean necesarios en el ciclo de vida del proyecto, ejemplo de las tareas de validación y verificación que son iterativas.

##### **1.1. Procesos de concertación:**

En esta sección se describen los procesos necesarios para establecer un punto de partida de suministros tanto recibidos como proporcionados por la organización.

###### **1.1.1. Proceso de adquisición:**

En este proceso se definen las acciones que debe realizar una organización para adquirir los productos y servicios necesarios para realizar un determinado proyecto.

###### **1.1.2. Proceso de suministrar:**

Acciones y tareas a realizar cuando se va a suministrar un producto o servicio.

##### **1.2. Procesos de iniciación de la organización del proyecto:**

En esta sección se describen los procesos necesarios para establecer el punto de partida de la planificación del proyecto y los recursos necesarios para realizarlo.

###### **1.2.1. Proceso de gestión de ciclo de vida:**

En este proceso se indican consideraciones para definir, mantener y asegurar la viabilidad de políticas, planificaciones del ciclo de vida, modelos del ciclo de vida y procedimientos usados por la organización para cumplir sus objetivos.

**1.2.2. Proceso de gestión de infraestructura:**

Acciones y tareas necesarias para establecer y mantener la infraestructura y servicios necesarios en el proyecto a lo largo de todos los procesos del ciclo de vida.

**1.2.3. Proceso de gestión de cartera del proyecto:**

El propósito de este proceso es el de iniciar y sostener los proyectos adecuados con el fin de cumplir con los objetivos de la organización. En este proceso se investigará la financiación y recursos adecuados para la realización del proyecto.

**1.2.4. Proceso de gestión de recursos humanos:**

En este proceso se definen las actividades necesarias para proveer a la organización de los recursos humanos necesarios para asegurar la realización y mantenimiento de las competencias y necesidades del negocio.

**1.2.5. Proceso de gestión de calidad:**

Este proceso señala las tareas y actividades a realizar para asegurar que los productos, servicios e implementaciones cumplen con los requisitos de calidad definidos por la organización y consiguen la satisfacción de los clientes.

**1.3. Procesos del proyecto:**

En esta sección se enumeran los procesos propios de la realización del proyecto.

**1.3.1. Proceso de planificación del proyecto:**

En este proceso se identifica el alcance de la gestión del proyecto estableciendo los hitos, entregables y actividades del proyecto. Establece los tiempos por cada tarea y los recursos necesarios para llevarla a cabo.

**1.3.2. Proceso de evaluación y control:**

El objetivo del proceso de evaluación y control es determinar que el estado del proyecto está al nivel adecuado de acuerdo a la planificación de tareas, tiempos de realización y presupuesto. En este proceso se determinan tareas para evaluar periódicamente el estado del proyecto y establece acciones para llevar a cabo en caso de detectar variaciones para orientar las replanificaciones.

**1.3.3. Proceso de gestión de decisiones:**

En este proceso se indican las acciones y tareas para llevar a cabo en caso de ser necesaria la toma de decisiones de gran importancia o impacto para el proyecto en cualquiera de sus etapas del ciclo de vida.

**1.3.4. Proceso de gestión de riesgos:**

En esta sección se describen tareas para identificar, analizar, monitorizar y resolver los riesgos de forma activa y continua a lo largo del ciclo de vida. Es un proceso sistemático que debe detectar y gestionar los riesgos en el mismo momento en el que se detectan.

**1.3.5. Proceso de gestión de configuraciones:**

En este proceso se definen las acciones a realizar para garantizar la integridad de los “outputs” o entregables del proyecto y hacerlos llegar a cada parte interesada.

**1.3.6. Proceso de gestión de información:**

Se describen las acciones a realizar para capturar, almacenar, gestionar y distribuir a cada parte interesada la información necesaria sobre el proyecto.

**1.3.7. Proceso de medición:**

En este proceso se detallan las tareas necesarias para realizar y analizar mediciones a los datos relativos a los productos desarrollados o procesos implementados dentro del proyecto. Estas mediciones deben apoyar y garantizar una gestión eficaz y demostrar objetivamente la calidad de los productos o servicios proporcionados.

**1.4. Procesos Técnicos:**

En esta sección se describen los procesos pertenecientes a la elaboración técnica del producto o servicio.

**1.4.1. Proceso de definición de requisitos de stakeholders:**

Se describe el proceso a seguir para definir los requisitos que surgen de las partes interesadas del proyecto o “stakeholders”.

El objetivo de realizar el proceso como indica la norma es conseguir especificar las características y el contexto de los servicios requeridos, definir las limitaciones de la una determinada solución para el sistema, definir la trazabilidad de los requisitos y las necesidades de las que surgen, definir las bases para establecer los requisitos de sistema y proporcionar una base de negociación para suministrar el producto o servicio a desarrollar.

- **Identificación de Stakeholders:**  
Se deberá identificar a cada participante o grupo de participantes interesados en el proyecto a lo largo de su ciclo de vida.
- **Identificación de requisitos:**  
Se describen las consideraciones que se deben tener en cuenta para realizar la obtención de los requisitos que parten de los stakeholders (cuál es la necesidad a cubrir, que es necesario realizar, como debe interactuar con otras partes del sistema, qué medidas de seguridad hay que tener en cuenta, definir su trazabilidad...etc.).
- **Evaluación de requisitos:**  
Se deben analizar todos los requisitos obtenidos en el paso anterior.
- **Acuerdo del requisito:**  
Se describen las consideraciones a llevar a cabo para acodar un requisito como válido.
- **Documentación del requisito:**  
Se describen las consideraciones a tener en cuenta para documentar y almacenar el requisito para su posterior utilización.

**1.4.2. Proceso de análisis de los requisitos:**

En este proceso se describen la transformación de los requisitos definidos por los stakeholders en un set de requisitos de sistema que servirán de guía para el diseño.

Como objetivos de realizar el proceso siguiendo la norma:

- Definir un set de requisitos funcionales y no funcionales que describan como resolver un problema como el stakeholder ha establecido.
- Las técnicas apropiadas son aplicadas para optimizar la solución del proyecto.
- Los requisitos del sistema son analizados y testeados de forma correcta.
- El impacto de cada requisito en el sistema es entendible.
- Los requisitos son priorizados y actualizados según necesidades.
- La consistencia y la trazabilidad entre requisitos queda establecida en la línea base.
- Los requisitos son comunicados y accesibles por todas las partes afectadas.

#### **1.4.3. Proceso de diseño de la arquitectura:**

En el proceso de diseño de la arquitectura se definen los métodos para disponer una arquitectura adecuada a las necesidades expresadas en los requisitos del sistema y acordes al presupuesto del proyecto teniendo en cuenta los riesgos de cada estrategia sobre el diseño arquitectónico.

#### **1.4.4. Proceso de implementación**

Este proceso describe las acciones necesarias para transformar el comportamiento específico, las interfaces y las limitaciones de la aplicación en acciones de fabricación que crean un elemento del sistema de acuerdo con las especificaciones de los requisitos. Este proceso resulta en un elemento del sistema que satisface las especificaciones de los requisitos de diseño a través de la verificación y validación.

#### **1.4.5. Proceso de integración:**

En este proceso se especifican las acciones a realizar para ensamblar o establecer un producto o servicio implementado en su arquitectura establecida. Este proceso combina elementos del sistema para establecer configuraciones completas o parciales con el fin de crear el sistema final.

#### **1.4.6. Proceso de verificación:**

El propósito del proceso de verificación es confirmar que los requisitos de diseño que se han especificado se cumplen en el sistema que ha sido implementado. Este proceso proporciona la información necesaria para llevar a cabo las acciones correctivas que permitan reelaborar las no conformidades en el sistema.

#### **1.4.7. Proceso de instalación:**

En este proceso se definen las acciones necesarias para la instalación e implantación de un sistema en su lugar de uso. Esto incluye las infraestructuras necesarias, procesos operativos, sistemas de soporte y todos los elementos necesarios especificados en los requisitos.

#### **1.4.8. Proceso de validación:**

El propósito del proceso de validación es establecer evidencia objetiva de que el sistema proporcionado cumple con las especificaciones de los requisitos y abarca todas las necesidades de los stakeholders que han sido documentadas y aceptadas.

**1.4.9. Proceso de operación:**

El objetivo de este proceso es utilizar el sistema con el fin de ofrecer sus servicios. Este proceso asigna personal para operar el sistema y controla los servicios y el rendimiento del operador. Con el fin de mantener los servicios se identifican y analizan los problemas de funcionamiento en función de acuerdos preestablecidos, los requisitos detectados y las limitaciones organizativas.

**1.4.10. Proceso de mantenimiento:**

La finalidad de las tareas descritas en el proceso de mantenimiento es mantener la capacidad del sistema para proporcionar sus servicios. Este proceso controla ésta capacidad y registra problemas para su análisis y corrección.

**1.4.11. Proceso de eliminación:**

El propósito del sistema de eliminación es poner fin a la existencia de una entidad del sistema. Este proceso desactiva, desmonta o elimina el sistema o parte del sistema, así como reciclar o retirar adecuadamente cualquier elemento perjudicial para el medio ambiente de acuerdo a las legislaciones activas.

### **2.3.4.5. ISO/IEC 12207 – Systems and software engineering – Software life cycle processes:**

La norma 12207 se publica como un estándar internacional que provee de un set de guías de aplicación en los procesos y actividades a realizar en cada etapa del ciclo de vida del desarrollo de productos software. En este estándar se desglosan todos los procesos de los que está, o puede estar constituido un proyecto de software desde su conceptualización hasta el mantenimiento del mismo una vez ya implantado. Este estándar ofrece una especialización de la norma ISO/IEC 15288 (2008) realizando una especialización para sistemas específicamente software.

En ella se dividen los procesos en dos bloques, “Ciclo de vida de los procesos de sistema” el cual se especifica en la norma ISO/IEC 15288(2008) y “Procesos específicos de software” que conforman los procesos específicos de software los cuales se describen a más bajo nivel que en la norma 15288.

#### **1. Ciclo de vida de los procesos de sistemas:**

Este bloque se compone de los mismos procesos que la norma ISO/IEC 15288(2008) los cuales están identificados en el apartado anterior “ISO/IEC 15288 – Systems and software engineering – Systems life cycle processes”.

#### **2. Procesos específicos de software:**

##### **2.1. Procesos de implementación de software:**

En esta sección del documento se describen las tareas y acciones que se recomiendan llevar a cabo para realizar la implementación de un producto software.

#### **2.1.1. Proceso de implementación software:**

Este proceso describe las acciones necesarias para transformar el comportamiento específico, las interfaces y las limitaciones de la aplicación en acciones de fabricación que crean un servicio o producto de software de acuerdo con las especificaciones de los requisitos. Este proceso resulta en un elemento de software que satisface las especificaciones de los requisitos de diseño a través de la verificación y validación.

#### **2.1.2. Proceso de análisis de los requisitos software:**

En este proceso se describen la transformación de los requisitos definidos por los stakeholders en un set de requisitos de software que servirán de guía para el diseño.

Como objetivos de realizar el proceso siguiendo la norma:

- Definir un set de requisitos funcionales y no funcionales que describan como resolver un problema como el stakeholder ha establecido.
- Las técnicas apropiadas son aplicadas para optimizar la solución del proyecto.
- Los requisitos del sistema son analizados y testeados de forma correcta.
- El impacto de cada requisito en el sistema es entendible.
- Los requisitos son priorizados y actualizados según necesidades.
- La consistencia y la trazabilidad entre requisitos queda establecida en la línea base.
- Los requisitos son comunicados y accesibles por todas las partes afectadas.

#### **2.1.3. Proceso de diseño de arquitectura software:**

En el proceso de diseño de la arquitectura se definen los métodos para disponer una arquitectura adecuada a las necesidades expresadas en los requisitos del software y acordes al presupuesto del proyecto teniendo en cuenta los riesgos de cada estrategia sobre el diseño arquitectónico. Los diseños realizados deben ser verificables en referencia a sus especificaciones descritas en los documentos de requisitos.

#### **2.1.4. Proceso de diseño detallado:**

Este proceso es una especificación a bajo nivel del proceso de implementación software. El propósito de este proceso es disponer de un diseño de software detallado con el cual se pueda implementar un determinado requisito de tal forma que sea posible su verificación, su correcta elaboración, su integración con la arquitectura definida y su adecuación a las especificaciones del documento de requisitos

#### **2.1.5. Proceso de construcción de software:**

Este proceso es una especificación a bajo nivel del proceso de implementación software. El propósito de este proceso es el de implementar unidades ejecutables que sean capaces de reflejar apropiadamente un diseño de software. Este ejecutable debe cumplir todos los criterios de verificación especificados en el documento de requisitos.

#### **2.1.6. Proceso de integración software:**

El objetivo de este proceso es combinar las unidades de software desarrolladas con sus componentes software, la integración de los diferentes elementos para completar un producto software productivo. Antes de integrar los elementos, cada uno de ellos debe



haber demostrado que cumple con las especificaciones reflejadas en los requisitos tanto funcionales como no funcionales y satisfacen enteramente las necesidades del cliente que fueron registradas.

#### **2.1.7. Proceso de test de calidad software:**

El propósito del proceso de test de calidad software es garantizar que el producto software que va a ser integrado cumple con los requisitos que han sido definidos. Se deberán realizar test de funcionalidad, rendimiento e integración hasta cumplir con los requisitos. Una vez pasada una auditoria para verificar que se han pasado los test de forma satisfactoria el software se considerará listo para ser integrado.

### **2.2. Procesos de soporte software:**

En esta sección se describen aquellos procesos que se desempeñan de forma análoga al desarrollo técnico de los elementos de software y que sirven para revisar, mejorar o ayudar a su realización.

#### **2.2.1. Proceso de gestión de la documentación software:**

Este proceso describe las tareas necesarias para desarrollar, gestionar y mantener la documentación producida a lo largo de cada etapa del ciclo de vida. Es de vital importancia documentar y almacenar cualquier información que implique una acción o modificación sobre el producto software que se está desarrollando.

#### **2.2.2. Proceso de gestión de la configuración software:**

En este proceso se describen las acciones necesarias para establecer y mantener la integridad de los elementos software y ponerlos a disposición de las partes interesadas.

#### **2.2.3. Proceso de aseguramiento de la calidad software:**

El objetivo de este proceso es asegurar que los elementos de software que son elaborados cumplen con las disposiciones y los planes predefinidos. El proceso de aseguramiento de la calidad se integra a su vez realizando de forma correcta las especificaciones descritas en los procesos de verificación, validación, revisión y auditoria de software y dando parte de que se han cumplimentado todas las especificaciones de calidad reflejadas en la planificación o contrato.

#### **2.2.4. Proceso de verificación software:**

El propósito del proceso de verificación es confirmar que los requisitos que han sido especificados se cumplen en el elemento de software desarrollado. Este proceso proporciona la información necesaria para llevar a cabo las acciones de verificación elaborando estrategias y planes de verificación para cada requisito.

#### **2.2.5. Proceso de validación software:**

El objetivo del proceso de validación de software es, básicamente, asegurar que los requisitos de un producto software han sido elaborados por completo. Existen requisitos que por su simplicidad no requieren un proceso de validación elaborado, pero para los requisitos más complejos esta sección de la normativa señala las acciones necesarias para elaborar un plan de validación para un componente software.

**2.2.6. Proceso de revisión software:**

En la sección del proceso de revisión de software se describen consideraciones a realizar con el objetivo de mantener un entendimiento común con los grupos interesados en el progreso de los objetivos de un determinado proyecto. Mediante este conocimiento común los elementos pueden ser revisados por profesionales de distintos perfiles y conocimientos que asegurarán que los requisitos satisfarán las necesidades de las partes interesadas. Las revisiones de software se realizarán a lo largo de la vida del proyecto tanto en su parte de desarrollo técnico como en las tareas de gestión del proyecto.

**2.2.7. Proceso de auditoría de software:**

Mediante el proceso de auditoría de software se pretende asegurar el correcto desarrollo de un producto de acuerdo a las especificaciones de sus requisitos y cumpliendo las normativas oportunas. Este proceso suele ser llevado a cabo por un equipo externo e independiente al equipo que desarrolla el proyecto para asegurar su imparcialidad y objetividad.

**2.2.8. Proceso de resolución de problemas software:**

En este proceso se describen las tareas necesarias para elaborar un protocolo de actuación frente a los problemas que se van detectando a lo largo del desarrollo de un producto. Mediante este protocolo el equipo será capaz de identificar, analizar, gestión y controlar la aparición y resolución de cualquier error o problema y comunicar el estado de los mismos a sus partes interesadas.

**2.3. Procesos de reutilización de software:**

Sección de procesos dedicada a la reutilización de dominios (planes, componentes, subsistemas, arquitecturas...etc.) entre los diferentes productos desarrollados por la organización.

**2.3.1. Proceso de ingeniería de dominio:**

El análisis del dominio es la identificación y explotación de componentes comunes en diferentes proyectos a través de un sistema relacionado para su futura reutilización en futuros desarrollos. En la descripción de este proceso se indican tareas a realizar para la identificación de dominios reutilizables y elaborar estrategias y planes de reutilización. Se considera un dominio reutilizable a componentes que definen un determinado tipo de arquitectura, vocabularios de definiciones, necesidades concretas de stakeholders, modelos y procesos de ingeniería de desarrollo.

**2.3.2. Proceso de reutilización de activos:**

El proceso de reutilización de activos muestra las acciones necesarias para identificar y gestionar la reutilización de determinados activos dentro de una misma organización para sus diferentes desarrollos. Para la reutilización de activos es necesario elaborar una estrategia de reutilización, realizar una clasificación de los activos reutilizables, realizar una gestión de los activos para controlar donde y en qué condiciones se están utilizando y elaborar un plan de almacenaje o mantenimiento de esos activos reutilizables mientras no están siendo utilizados por ningún proceso de desarrollo.

**2.3.3. Proceso de programa de reutilización:**

El proceso de programa de reutilización consta de los requerimientos necesarios para establecer, administrar, controlar y supervisar un programa de reutilización de recursos dentro de una organización. Se deben identificar todos los posibles dominios de reutilización así como el alcance y potencial reutilizable de cada dominio. Se elaborarán propuestas y estrategias de reutilización para proveer a la organización de mecanismos de retroalimentación, comunicación y notificación tales que permitan identificar cuando un dominio previamente detectado puede ser reutilizado para una nueva necesidad.

### 2.3.5. Herramientas de creación y gestión de requisitos:

En el espacio actual del mercado de herramientas de software que apoyan a los ingenieros de sistemas en el desarrollo de los diferentes componentes y artefactos que componen un producto para un sistema específico y a raíz de la cada vez mayor importancia que los requisitos han ido obteniendo dentro del desarrollo de proyectos existen a disposición de los desarrolladores y corporaciones una extensa, y a veces confusa, oferta de diferentes herramientas para la creación, almacenamiento y gestión de los requisitos.

Estas herramientas ofrecen a sus usuarios un gran abanico de utilidades para facilitar la realización de las múltiples tareas relacionadas con la ingeniería de requisitos y por lo general tienen una amplia curva de aprendizaje debido a su alcance y complejidad. La tendencia actual de las empresas desarrolladoras de estas herramientas es disminuir esta complejidad y realizar productos fáciles e intuitivos sin comprometer su calidad.

Para este proyecto de fin de carrera se han elegido 3 de las múltiples herramientas disponibles en el mercado. Se han escogido estas herramientas porque cada una facilita la gestión de la calidad de los requisitos desde un enfoque diferente.

Una de estas herramientas es **Innoslate**, la cual ofrece utilidades de análisis semántico sobre los requisitos para determinar su calidad además de otras funcionalidades que facilitan a los desarrolladores realizar el trabajo con los requisitos.

Otra de las herramientas elegidas para este trabajo es **Jama Product Delivery Platform**. Se ha elegido esta herramienta porque ofrece a sus usuarios una gran conectividad entre ellos y pone un énfasis particular en el proceso de revisión de los requisitos, al final del cual cada requisito es revisado y aceptado cuando alcanza una calidad determinada.

La última de las herramientas elegidas para este trabajo es **Requirements Quality Analyzer**, herramienta que forma parte de la suite de herramientas **Requirements Quality Suite**. Se ha elegido esta herramienta porque utiliza como solución innovadora el uso de repositorios de conocimiento y análisis semántico sobre el lenguaje como métodos para la generación y el análisis de requisitos.

### 2.3.5.1. Innoslate - Innovaciones SPEC ®:

Innovaciones SPEC es una empresa que se encuentra en proceso de desarrollar un conjunto de herramientas basadas en la idea de una aplicación de gestión de ingeniería de sistemas funcional que puede proporcionar todo lo que pueda necesitar el ingeniero de sistemas para desarrollar y mantener los artefactos de un producto de trabajo en un lugar de fácil acceso y centralizado, accesible en cualquier momento y desde cualquier lugar.

Innoslate es una aplicación web basada en la nube que Innovaciones SPEC está desarrollando basándose en esa premisa de aplicación de sistemas integrales utilizando el modelo de software como servicio (SaaS: Software-as-a-Service).

Innoslate es el primer producto de software comercial de la compañía, ya que esta se centraba en desarrollos de productos basados en el marco de arquitectura (DoDAF - Department of Defense Architecture Framework) y abarca una aplicación web basada en la nube que ofrece una gama de capacidades para gestionar todos los aspectos de la ingeniería de sistemas y del ciclo de vida de desarrollo desde la visión de los requisitos y sus derivaciones en componentes del producto. Innoslate fue concebido como la agregación de varias herramientas y conceptos diferentes utilizados en la captura y desarrollo de elementos de ingeniería, requisitos, modelos y diseños.

#### 2.3.5.1.1. **Funcionalidades principales:**

Como se comentaba anteriormente Innoslate se compone de la agregación de varias herramientas lo cual le dota de múltiples funcionalidades cuyas principales se describen a continuación:

##### 2.3.5.1.1.1. **Creación de elementos de información en la BBDD:**

Innoslate permite la creación, modificación y borrado de elementos (entities) los cuales almacena en su base de datos y conforman los elementos de información necesarios para la gestión de los requisitos de un proyecto.

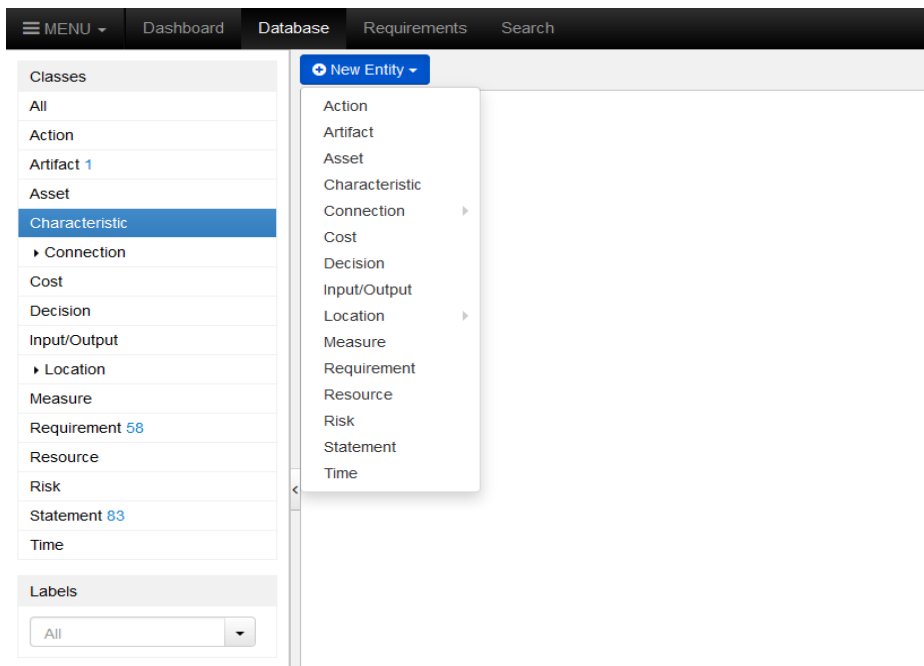


Figura 16: Pantalla de creación de entidades de Innoslate.

Estos elementos son de diferentes tipos:

- Actions: Son tareas o trabajos a realizar.

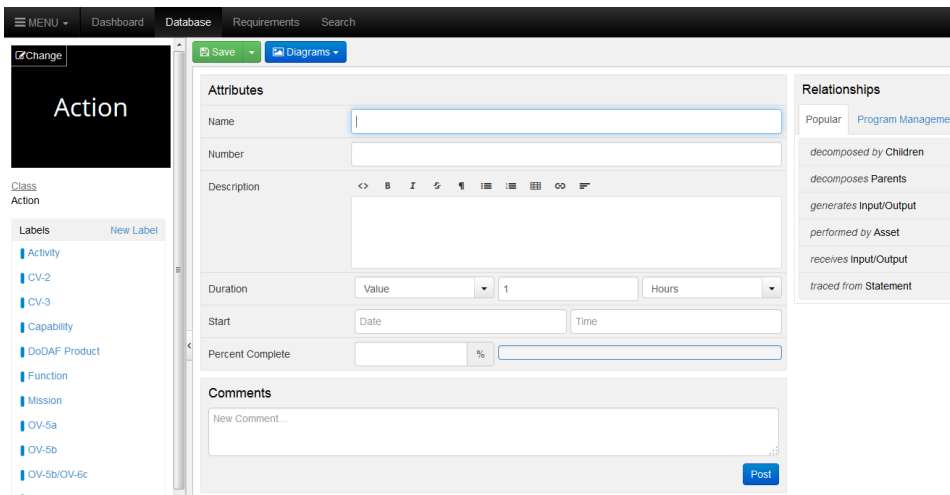


Figura 17: Pantalla de creación de “Actions” de Innoslate.

- Artifacts: Son elementos externos de información, documentos de texto, hojas de cálculo, ficheros de E/S o cualquier archivo o documento necesario.



Figura 18: Pantalla de creación de “Artifacts” de Innoslate.

- Assets: Son bienes o activos utilizados en el proyecto.

Figura 19: Pantalla de creación de “Assets” de Innoslate.

- Characteristics: Son atributos aplicables a requisitos, se define el atributo, una unidad para cuantificarlo y el valor de esa unidad.



Navigation: MENU | Dashboard | Database | Requirements | Search

Buttons: Save | Diagrams

Change

Characteristic

Class  
Characteristic

Labels [New Label](#)

- Condition
- DoDAF Product
- Final State
- Initial State
- State

Auto Managed Labels

Attributes

Name:

Number:

Description: 

<> B I S

Value:

Units:

Comments

New Comment...

Post

Relationships

Popular

decom

decom

specifi

Figura 20: Pantalla de creación de “Characteristics” de Innoslate.

- Connections: Representan las transacciones de datos o elementos entre los componentes del proyecto, pueden ser de tipo “Conduit” o “Logical”.

Navigation: MENU | Dashboard | Database | Requirements | Search

Buttons: Save | Diagrams

Change

Conduit

Class  
Connection/ Conduit

Labels [New Label](#)

- Aggregation
- Association
- Cable
- Composition
- DoDAF Product
- Downlink
- Interface
- Network
- Pipe
- Roadway

Attributes

Name:

Number:

Description: 

<> B I S

Capacity: Value  0

Latency: Value  Number  Seconds

Units:

Comments

New Comment...

Post

Relationships

Popular

Prog

connects to As

decomposed b

decomposes P

transfers Input

Figura 21: Pantalla de creación de “Conections” de tipo “Conduit” de Innoslate.



Figura 22: Pantalla de creación de “Conections” de tipo “Logical” de Innoslate.

- Costs: Son los costes asociados al proyecto o a determinados elementos del proyecto. Se pueden expresar como costes fijos o variables en el tiempo.

Figura 23: Pantalla de creación de “Costs” de Innoslate.

- Decisions: Son decisiones ligadas a los diferentes componentes del proyecto, las cuales quedan registradas y almacenadas.





Figura 24: Pantalla de creación de “Decisions” de Innoslate.

- Inputs/Outputs: Diferentes entradas y salidas del proyecto las cuales son descritas y tipificadas.

Figura 25: Pantalla de creación de “Inputs/Outputs” de Innoslate.

- Locations: Son elementos de localización los cuales pueden ser orbitales (Orbital), direcciones físicas (Physical) o direcciones virtuales (Virtual).



Orbital

Class Location/ Orbital

Labels New Label

DoDAF Product

Auto Managed Labels

Name

Number

Description

Apoapsis Value 0

Periapsis Value 0

Eccentricity Value 0

Semimajor Axis Value 0

Inclination Value 0

Reference Plane

Longitude of Ascending Node Value 0

Origin of Longitude

Argument of Periapsis Value 0

Mean Anomaly Value 0

Comments

New Comment...

Popular Program Management All

decomposed by Children

decomposes Parents

locates Many

Figura 26: Pantalla de creación de “Locations” de tipo “Orbital” de Innoslate.

Physical

Class Location/ Physical

Labels New Label

DoDAF Product

Fixed

Mobile

Moveable

Auto Managed Labels

Attributes

Name

Number

Description

Address

Coordinates Latitude Longitude Select

Altitude Value 0

Comments

New Comment...

Post

Relationships

Popular Program Management All

decomposed by Children

decomposes Parents

locates Many

Figura 27: Pantalla de creación de “Locations” de tipo “Physical” de Innoslate.



Figura 28: Pantalla de creación de “Locations” de tipo “Virtual” de Innoslate.

- Measures: Mediciones aplicables sobre los distintos componentes del proyecto.

Figura 29: Pantalla de creación de “Measures” de Innoslate.

- Requirements: Estas son las entidades centrales las cuales conforman los diferentes requisitos del sistema y son relacionadas al resto de entidades. Cada requisito tiene un código identificador, nombre, descripción y la necesidad o justificación de la existencia del requisito. Innoslate incorpora un control en la calidad de los requisitos (Quality Score), el cual se expondrá más adelante, cuya puntuación se calcula a través de una serie de criterios de calidad (clear, complete, consistent, correct, desing, feasible, modular, traceable, verifiable) algunos de los cuales se analizan automáticamente mediante una herramienta de análisis sintáctico sobre la descripción del requisito y otros deben ser asignados por los revisores del requisito. Este control de la calidad del requisito da una gran



visibilidad sobre su estado y madurez. Adicionalmente a cada requisito se le puede relacionar con otros elementos de Innoslate, ya sean otros requisitos, documentos (artifacts), elementos de entrada/salida (input/output) o de cualquier otro tipo disponible.

Save

Diagrams

No other viewers

Attributes

Name

Number

Description

Quality Score

0%

Rationale

Clear

No

Complete

No

Consistent

No

Correct

No

Design

No

Feasible

No

Modular

No

Traceable

No

Verifiable

No

Comments

New Comment...

Post

Relationships

Popular

Program Management

All

decomposed by Children

Add

decomposes Parents

Add

references Artifact

Add

sourced by Artifact

Add

traced from Statement

Add

traced to Many

Add

Figura 30: Pantalla de creación de requisitos de Innoslate.

- Resources: Representan los recursos (humanos o materiales) de los que puede constar el proyecto.



Figura 31: Pantalla de creación de “Resources” de Innoslate.

- **Risks:** Entidades en las que se almacenan los distintos riesgos que surgen en el proyecto, ya sean por no abordar una tarea a tiempo, riegos implícitos dentro de la realización de tareas determinadas. Estos riesgos son identificados, almacenados y están relacionados con otros elementos de Innoslate como requisitos, tareas....etc.

Figura 32: Pantalla de creación de “Risks” de Innoslate.

- **Statements:** Son declaraciones que representan sirven para identificar y agrupar un conjunto o subconjunto de elementos dentro del proyecto. En la vista de requisitos (descrita más adelante) los statements agrupan requisitos para representar una sección o etapa determinada compuesta por los requisitos del conjunto.

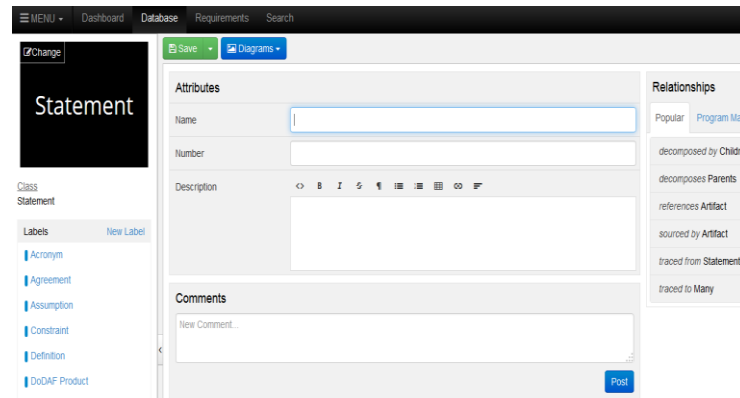


Figura 33: Pantalla de creación de “Statements” de Innoslate.

- Time: Establecen periodos temporales, los cuales son normalmente usados para marcar las duraciones de las diferentes tareas a realizar en el proyecto.

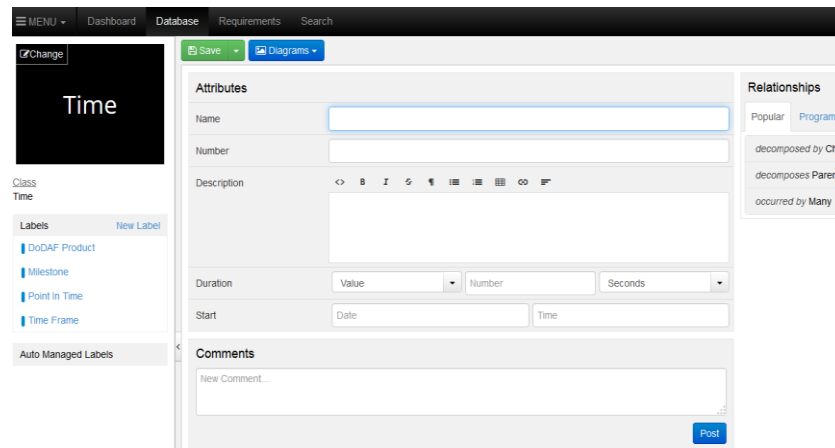


Figura 34: Pantalla de creación de “Time” de Innoslate.

#### 2.3.5.1.1.2. Panel de Control de Innoslate “Dashboard”:

El panel de control de Innoslate o “dashboard” es la pantalla principal de la aplicación la cual se muestra tras hacer el login. Como elementos principales en esta pantalla nos encontramos fundamentalmente tres: un cuadro de notificaciones, una representación gráfica del modelo de madurez y un gráfico de estadísticas del número y tipo de los elementos de la base de datos del proyecto.

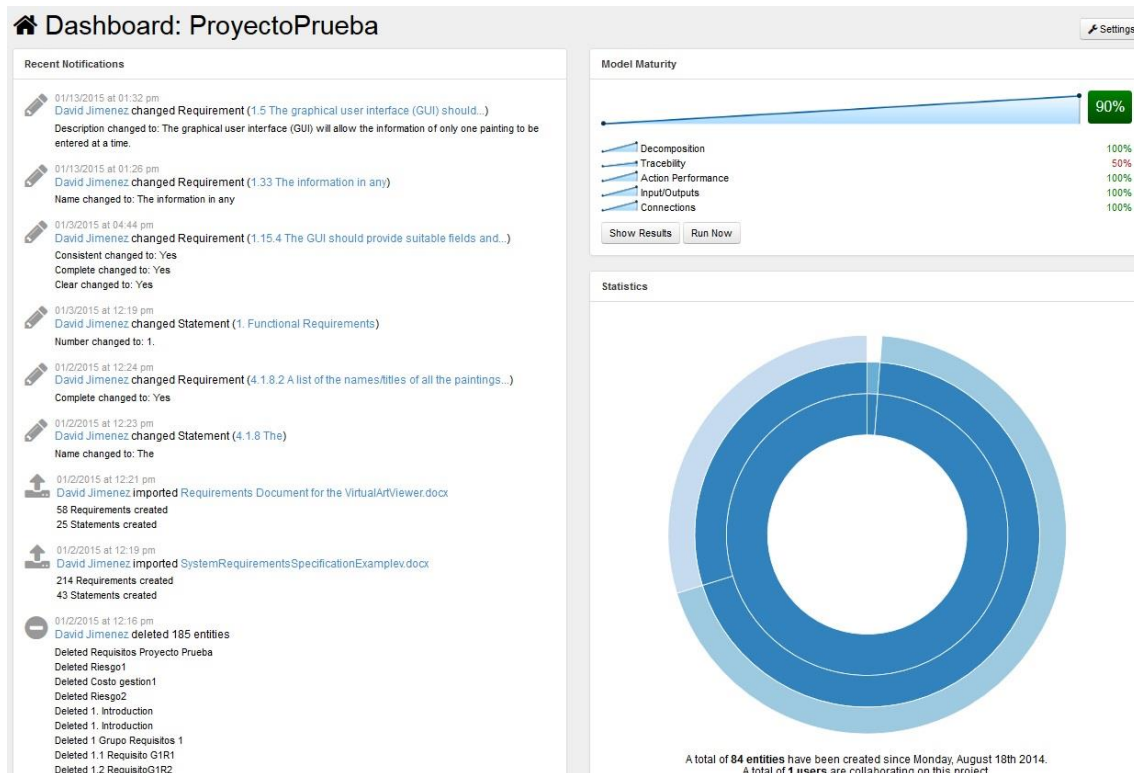


Figura 35: Panel de Control “Dashboard” de Innoslate.

- **Cuadro de notificaciones:** En el cuadro de notificaciones podemos ver un histórico de todas las acciones realizadas sobre el proyecto con información sobre qué elemento se ha cambiado, que atributo ha sido añadido o modificado en el elemento, cuando se ha realizado la acción y por quién ha sido llevada a cabo. Esto permite un control y una visión inicial e inmediata sobre los elementos en los que se está trabajando en la actualidad y quién está realizando ese trabajo en el momento.
- **Modelo de madurez:** En este gráfico se muestra la madurez de los elementos creados en la base de datos indicando los tipos de elementos y su porcentaje de madurez.
- **Estadísticas:** Ofrece una visión general del número de elementos creados en el proyecto y su proporción respecto a los diferentes tipos de elementos.

#### 2.3.5.1.1.3. **Menú de Requisitos:**

Innoslate provee de una visión generalizada de todos los requisitos contenidos en un documento específico. Estos documentos se pueden importar desde otras aplicaciones como Rational DOORS o Microsoft Word y exportar desde el propio Innoslate.

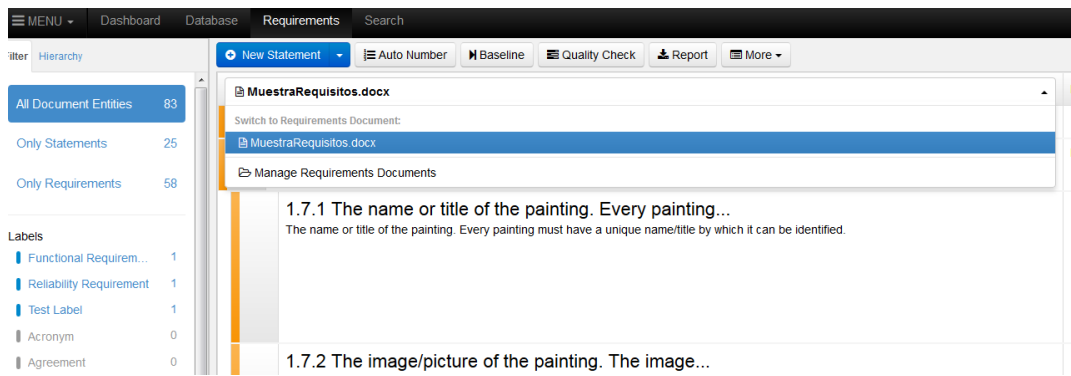


Figura 36: Selección de documentos de requisitos.

	Rationale	Quality Score	Labels	Clear	Complete
1.1 Functional Requirements ADMINISTRATOR FUNCTIONALITY Administrator Login	N/A	N/A	None to display	N/A	N/A
1.2 All Apogee Arts, Inc. administrators have... All Apogee Arts, Inc. administrators have to be verified using a login and password that is issued by Apogee Arts, Inc. The application should offer the administrator a set of user interfaces different from those offered to the normal user since he has access to features that allow him to edit, delete and add information to the database. Creating New Paintings		11%	Test Label	No • Contains a conjunction; consider separating into two statements.	No • Requirement contains more than one sentence. • You may need to reformat the Inc. acronym.
1.3 Every painting always belongs to only one... Every painting always belongs to only one "Theme" for example such as modern art or nature, etc.	N/A	N/A	None to display	N/A	N/A
1.4 The system has a drop-down list of themes... The system has a drop-down list of themes already created for an administrator to choose from while adding a new painting information.	N/A	N/A	None to display	N/A	N/A
1.5 The graphical user interface (GUI) should... The graphical user interface (GUI) will allow the information of only one painting to be entered at a time.		44%	None to display	Yes	Yes
1.6 The GUI should allow the administrator to create... The GUI should allow the administrator to create new painting information, retrieve and edit existing painting information and save them. If painting information section with the same title already exists, the GUI should prompt administrator accordingly.		11%	None to display	No • Contains a conjunction; consider separating into two statements.	No • Requirement contains more than one sentence.
1.7 The The "Painting information" for a painting includes:	N/A	N/A	None to display	N/A	N/A
1.8 The GUI should allow the administrator to create... The GUI should allow the administrator to create new artist information, retrieve and edit existing artist information and save it. If an artist information section with the same name of the artist already exists, the GUI should prompt administrator accordingly.		11%	Functional Requir... Usability Requir...	No • Contains a conjunction; consider separating into two statements.	No • Requirement contains more than one sentence.
1.9 The	N/A	N/A	None to display	N/A	N/A

Figura 37: Vista general de los requisitos.

Dentro de cada documento se presenta una visión de todos los requisitos que lo componen de forma jerarquizada mostrando así sus dependencias. Tal y como se muestra en la figura anterior en un primer vistazo podemos observar el título del requisito y una muestra del contenido de su descripción, además Innoslate permite configurar la información que necesitamos mostrar en este panel ofreciendo múltiples configuraciones según los intereses del proyecto. En la figura 34se puede observar, por ejemplo, la justificación del requisito en la columna “Rationale”, una puntuación sobre la calidad (de la cual se hablará más adelante) en la columna “Quality Score”, una columna “Labels” con las etiquetas que podremos asignar a cada requisito la cuales resultan muy útiles para identificar determinados atributos y para realizar posteriormente búsquedas de forma más óptima. Innoslate proporciona una serie de columnas de información preconfiguradas para esta ventana pero es posible crear columnas nuevas o añadir o quitar elementos de información según convenga.

Otros elementos de interés en esta ventana son el botón “Baseline” el cual establece la línea base de los requisitos comprendidos en el documento seleccionado, el botón



“Report” con el que se exportará un documento con la información de los requisitos, y el botón “Quality Check” el cual se explicará más adelante.

#### 2.3.5.1.1.4. **Generación de Diagramas:**

Innoslate permite generar de forma automática varios tipos de diagramas que muestran su información almacenada de forma gráfica facilitando la visualización de los conjuntos de contenido. Entre estos tipos destacan los diagramas de árbol, de jerarquía, el diagrama de araña, el timeline y el diagrama de trazabilidad.

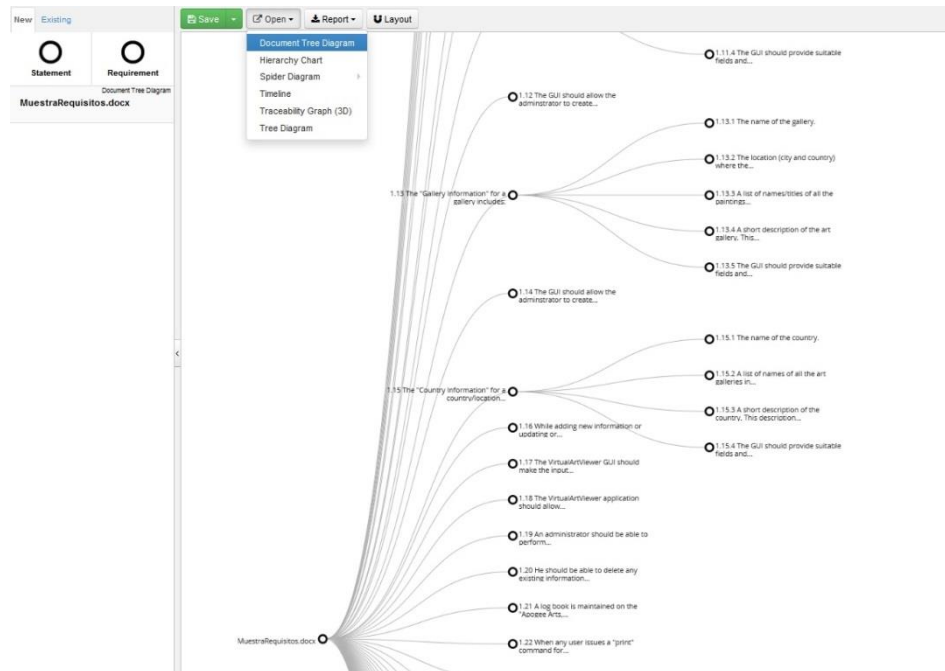


Figura 38: Ejemplo diagrama de tipo árbol.

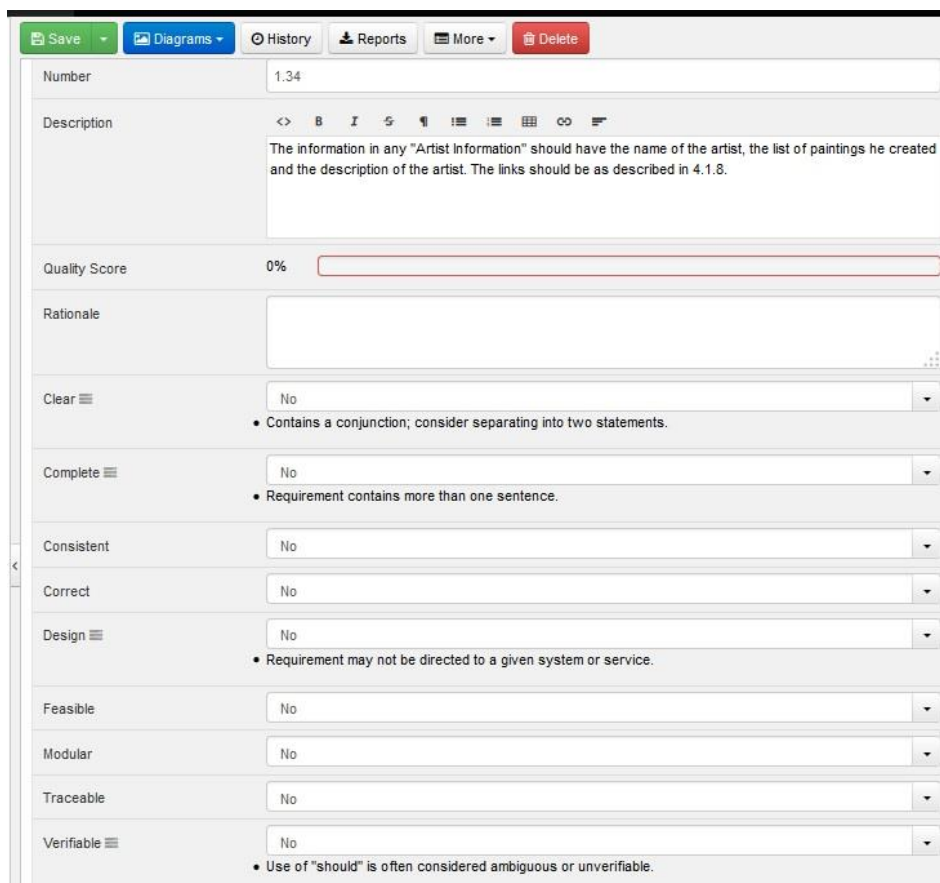
#### 2.3.5.1.2. **Gestión de la calidad en Innoslate:**

Una de las principales características de Innoslate, y razón por la cual se ha decidido incluir en este proyecto, es el enfoque que ofrece a la gestión de la calidad de sus requisitos utilizando diferentes técnicas y herramientas dentro de sus opciones.

Una vez redactado un requisito Innoslate nos permite realizar un análisis semántico automático sobre los contenidos que se han añadido, mediante el botón “Quality Check” el cual puede observarse en la figura 36. Esta función realizará un análisis sobre todos los requisitos contenidos en el documento de especificación abierto en ese momento y mostrando una puntuación sobre el nivel de su calidad en el atributo “Quality Score” el cual también es mostrado en la figura 36. Esta puntuación se consigue en base al cumplimiento de las características necesarias de un requisito bien formulado, es decir, si son claros, completos, consistentes, factibles, realizables, modulares, trazables y verificables.

Innoslate es capaz, mediante un análisis semántico de la descripción de los requisitos, de evaluar de forma automática algunos de estos atributos utilizando diferentes métricas como uso de expresiones típicas, utilización de la voz activa o pasiva, uso determinadas

formas verbales (uso de tiempos condicionales, restricciones sobre verbos ambiguos..etc.) entre otras técnicas ofreciendo avisos para su corrección tal y como se observa en la siguiente figura.



The screenshot displays a web-based interface for managing requirements. At the top, there is a navigation bar with buttons for 'Save', 'Diagrams', 'History', 'Reports', 'More', and 'Delete'. Below this, the main form is divided into several sections:

- Number:** A text field containing '1.34'.
- Description:** A rich text editor containing the text: 'The information in any "Artist Information" should have the name of the artist, the list of paintings he created and the description of the artist. The links should be as described in 4.1.8.'
- Quality Score:** A progress bar showing '0%'.
- Rationale:** A large text area for providing justification.
- Clear:** A dropdown menu set to 'No', with a tooltip: 'Contains a conjunction; consider separating into two statements.'
- Complete:** A dropdown menu set to 'No', with a tooltip: 'Requirement contains more than one sentence.'
- Consistent:** A dropdown menu set to 'No'.
- Correct:** A dropdown menu set to 'No'.
- Design:** A dropdown menu set to 'No', with a tooltip: 'Requirement may not be directed to a given system or service.'
- Feasible:** A dropdown menu set to 'No'.
- Modular:** A dropdown menu set to 'No'.
- Traceable:** A dropdown menu set to 'No'.
- Verifiable:** A dropdown menu set to 'No', with a tooltip: 'Use of "should" is often considered ambiguous or unverifiable.'

Figura 39: Sugerencias tras análisis de un requisito.

Otros de estos atributos deben ser analizados y calificados por un usuario que ejerza de revisor, el cual será al final de su revisión, el que garantice la calidad de cada requisito.

Actualmente los desarrollos de un mismo proyecto pueden ser realizados por diferentes empresas o incluso por una misma empresa con sus trabajadores diseminados en distintas ubicaciones. Innoslate al ser una herramienta en la nube facilita la comunicación y la transmisión de información entre sus usuarios quienes pueden comunicarse directamente mediante su chat interno ofreciendo la posibilidad de mandar un mensaje a todos los trabajadores que estén conectados en ese momento, mandar un mensaje a un determinado usuario o elaborar grupos cerrados de usuarios.

Por otro lado, la herramienta sigue un estricto control sobre las modificaciones realizadas sobre sus componentes ofreciendo una completa trazabilidad sobre qué elementos se han modificado, sobre qué atributos se ha trabajado, qué usuario ha realizado la modificación y cuando ha sido realizada.

#### **2.3.5.1.3. Conclusiones sobre Innoslate:**

Innoslate al ser un compendio de varias herramientas ofrece gran variedad de utilidades para el desarrollo de los requisitos de un proyecto. Almacena la información de forma efectiva y totalmente configurable pudiendo añadir las características necesarias sobre uno de sus componentes que la aplicación no ofreciera por defecto.

Dispone de una gran variedad de opciones para compatibilizarse con otras herramientas de gestión de requisitos, pudiendo importar, exportar y analizar documentación con diferentes formatos (excel, csv, DOORS...etc.).

Innoslate es fácil de utilizar ya que su manejo es muy visual y el acceso a sus distintas utilidades es muy directo. Innoslate es totalmente gratis para proyectos de pequeño tamaño y dispone de multitud de manuales y videoconferencias (webinars) de acceso libre.

Es una herramienta en la nube, accesible vía web desde cualquier navegador. Esto permite tener la información disponible desde cualquier equipo y en cualquier momento, no obstante es imprescindible tener una conexión a internet constante y de una potencia media y depender de un servidor de terceros para acceder al total de la información de tu proyecto puesto que es almacenada de forma externa.

Por todo esto Innoslate es una herramienta muy útil si se usa junto a otras herramientas de requisitos. Innovaciones SPEC sigue trabajando y mejorando su herramienta ofreciendo más utilidades y refinando las que ya dispone, lo cual puede llegar a convertirle en un competidor de las herramientas de referencia en el mercado.

### **2.3.5.2. Jama Product Delivery Platform - Jama Software:**

Jama Software es una empresa fundada en el año 2007 con sede en Portland, Oregón. Su principal producto es “*Jama Product Delivery Platform*” (al que abreviaremos como “Jama” en este trabajo) una aplicación web colaborativa basada en la nube que ayuda a las compañías en las labores de la gestión de sus proyectos y los requisitos de éstos.

En el año 2011 apareció en la lista Forbes de las empresas más prometedoras en la posición número 52 según [23] y actualmente goza de una gran trayectoria comercial.

Jama está diseñada con la intención de conectar a todos los equipos o participantes de un mismo proyecto de una forma totalmente fluida y colaborativa, donde la información es fácilmente visible y manejable por todos los integrantes del proyecto.

Tanto la herramienta como toda la documentación necesaria para utilizarla están disponibles en [24].

#### **2.3.5.2.1. Funcionalidades principales:**

Aunque se trata de una herramienta web, Jama tiene una visualización muy similar a los entornos de desarrollo (IDEs) más utilizados por los desarrolladores (Eclipse, IntelliJ IDEA, NetBeans...etc.) con ello la mayoría de sus usuarios se familiarizarán muy rápidamente a la herramienta accediendo a sus múltiples utilidades de forma intuitiva. Jama permite la migración de sus contenidos con otras herramientas similares como DOORS y con documentos Office que sigan ciertos patrones, mediante la exportación/importación de archivos. También está preparada para integrarse con otras herramientas de desarrollo como JIRA, Rally, TFS y HPQC.

A continuación se expone la funcionalidad de sus ventanas principales:

##### **2.3.5.2.1.1. Pantalla inicial (Dashboard) :**

Una vez realizado el login la aplicación muestra por defecto su pantalla principal. En ella se muestran 3 bloques: los últimos proyectos en los que el usuario ha estado trabajando, las revisiones que el usuario tiene pendientes en ese momento y los hilos de conversación activos. Estos bloques son una visión general de la actividad que se está realizando en el momento y permite un primer vistazo al estado actual del proyecto.

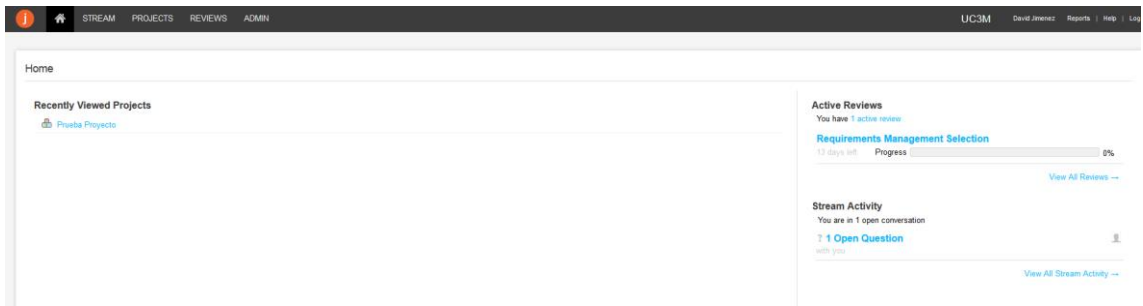


Figura 40: Página de Inicio de Jama.

2.3.5.2.1.2. **STREAM (Comentarios) :**

La utilidad de STREAM permite a los usuarios comunicarse entre ellos de forma similar a las actuales herramientas de microblogging. Mediante mensajes de texto cortos, utilización de menciones a otros usuarios o grupos de usuarios, utilización de hashtags y tipificación de los comentarios pudiendo remarcarlos como preguntas, errores y toma de decisiones se ofrece un medio de comunicación colaborativo entre los componentes del proyecto. Cada mención a un usuario, o los comentarios marcados como error, pregunta o decisión aparecerán en su página principal.

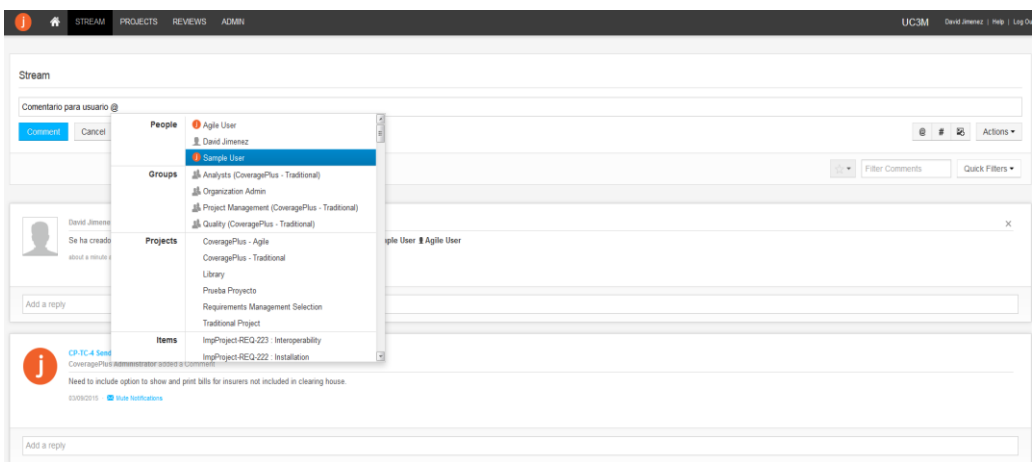


Figura 41: Comentario en ventana de STREAM.

2.3.5.2.1.3. **Projects (Gestión de proyectos) :**

La ventana para la gestión de proyectos alberga la mayor parte de la funcionalidad principal de Jama. En ella se concentran todas las utilidades necesarias para la gestión de proyectos y la creación y manejo de todos los elementos que los conforman.

Cada proyecto está compuesto de los tipos de componentes que son habitualmente manejados por las organizaciones en sus desarrollos, cada uno de estos elementos consta de un identificador propio, nombre y descripción. Dependiendo del tipo tienen otros atributos de información predeterminados y todos ellos son editables para adaptarse a las necesidades de los desarrolladores.

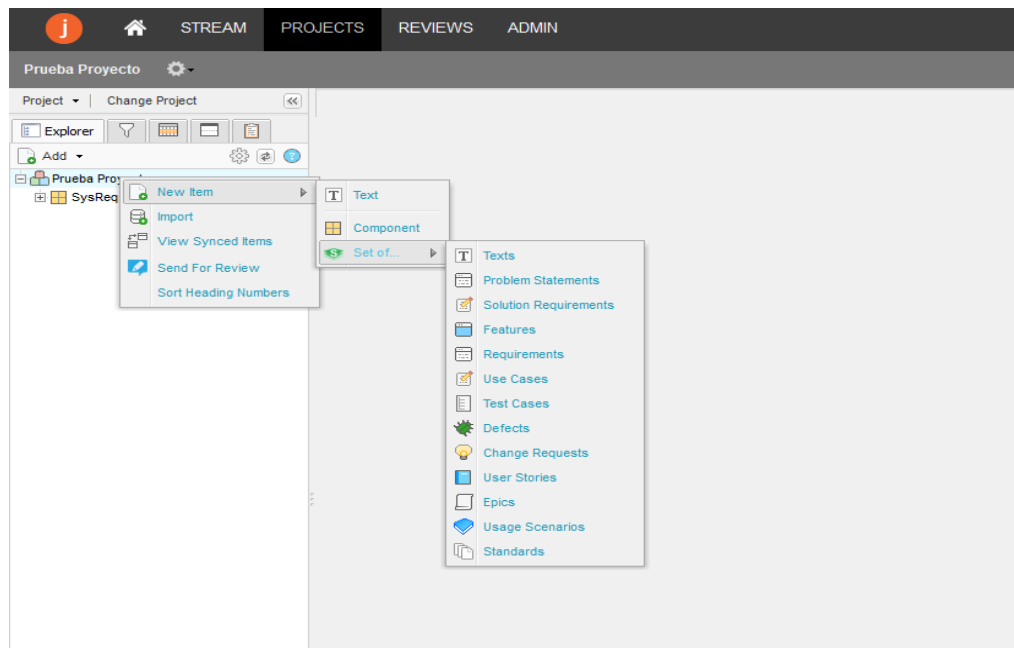


Figura 42: Creación de elementos en Jama

Los elementos que Jama pone a disposición son:

- Características (*features*): Documentos que representan las características generales del producto y no tienen cabida en ningún otro tipo de elemento.
- Requisitos (*requirements*): Requisitos del producto. Cada requisito consta de su identificador, nombre, descripción, prioridad, usuarios asignados al requisito, su estatus actual y a que release pertenecen.
- Casos de uso (*use cases*): Inventario de casos de uso detectados. Jama permite almacenar imágenes, diagramas para ilustrar estos casos de uso.
- Tests (*tests cases*): Tests para cada requisito.
- Historias de usuario (*user stories*): Historias de usuario inventariadas de las cuales se generan los requisitos.
- Estándares y normativas (*standards*): Normativas y estándares que se aplican en el desarrollo del producto.
- Problemas detectados (*problema statements*): Problemas funcionales que han sido detectados para su tratamiento.
- Requisitos externalizados (*solution requirements*): Requisitos cuyo solución será desarrollada por una organización diferente.
- Errores (*defects*): Errores sobre los componentes elaborados.
- Peticiones de cambios (*change requests*): Peticiones de cambios de los requisitos a la espera de ser estudiadas, validadas y aplicadas.
- Textos (*texts*): Textos de apoyo documental para cualquiera de los componentes del proyecto.
- Escenarios de uso (*usage scenarios*): Escenarios de uso que ilustrarán las necesidades del usuario frente a los requisitos.

Cada elemento consta de utilidades adicionales tales como una caja de comentarios propia de cada ítem, un listado de actividades realizadas sobre el ítem en las que se muestran qué y quién ha editado, los elementos adicionales que tiene conectados, las relaciones con otros elementos, ítems con los que está sincronizado, las etiquetas (tags) que caracterizan al elemento y un histórico de versiones.

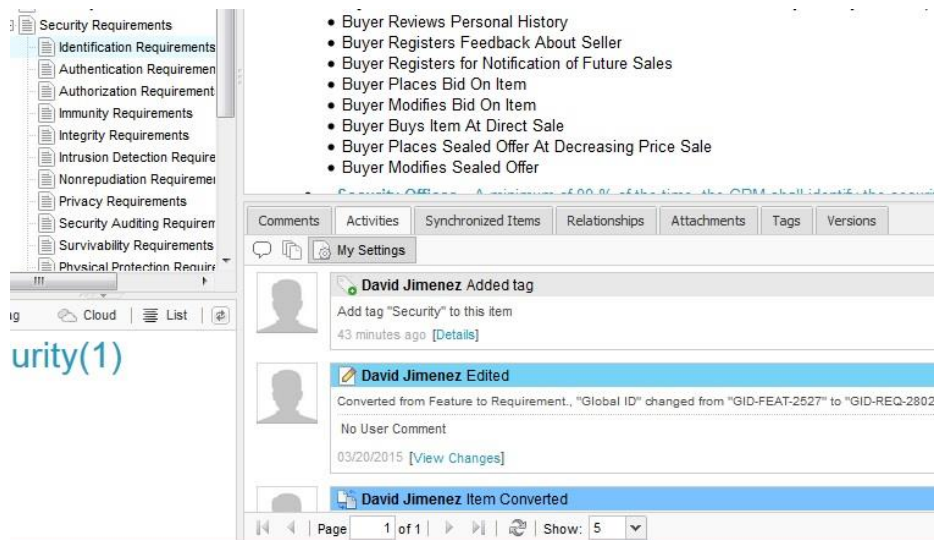


Figura 43: Cuadros de utilidades de los elementos

#### 2.3.5.2.1.4. Reviews (Revisiones) :

La ventana para la realización de revisiones es una de las funcionalidades más destacadas de Jama. Esta utilidad facilita en gran medida la realización del proceso de revisión sin necesidad de que los participantes estén en el mismo lugar o en el mismo momento trabajando en ella.

El proceso de revisión parte de un usuario, que ejercerá el rol de moderador de la revisión, el cual elige que requisitos que formarán parte de la revisión, durante cuánto tiempo estará vigente el proceso y quienes son los usuarios que ejercerán el rol de revisores y aprobadores, después envía la revisión mediante una convocatoria que será recibida por todos sus participantes.

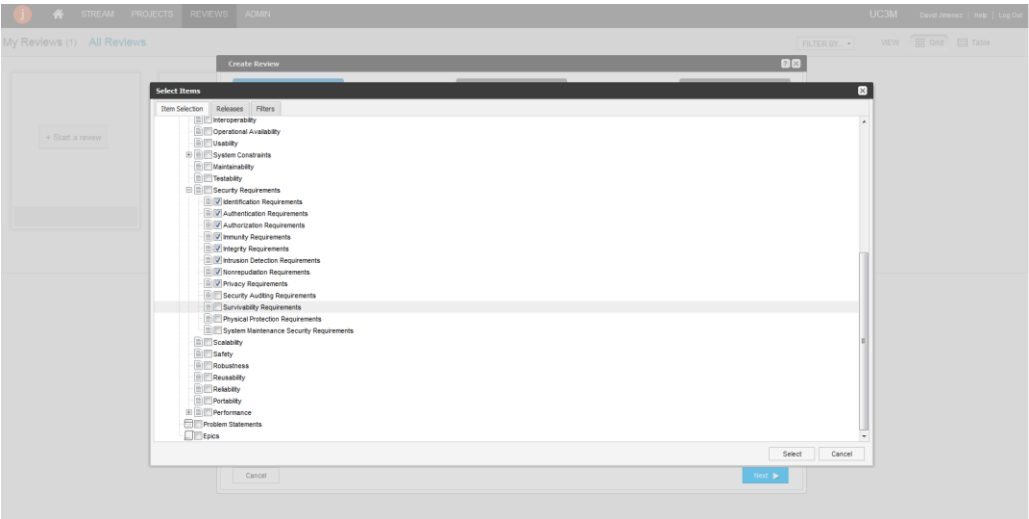


Figura 44: Selección de requisitos para revisión

A los revisores y aprobadores les aparecerá un aviso en la ventana principal así como un email avisándoles del periodo de revisión que ha sido abierto.

El usuario con rol de revisor podrá acceder a la revisión donde, mediante una lectura de los requisitos asociados, podrá realizar comentarios sobre el requisito o partes del requisito y marcarlos como revisados una vez no tenga nada más que aportar en él.

El usuario con rol de aprobador podrán marcar los requisitos una vez revisados como “necesitan más trabajo” (etiqueta 1 en la siguiente figura) o “aprobados” (etiqueta 2 en la siguiente figura) y añadir comentarios sobre ellos (etiqueta 3 en la siguiente figura) o comentarios sobre ciertas secciones del requisito (etiqueta 4 en la siguiente figura). El moderador se encargará de editar los requisitos en base a los comentarios aportados por los revisores y aprobadores en caso de que el requisito no esté completado. Si un requisito es modificado durante el periodo de revisión esta modificación será notificada a los revisores para que vuelvan a revisar el requisito.

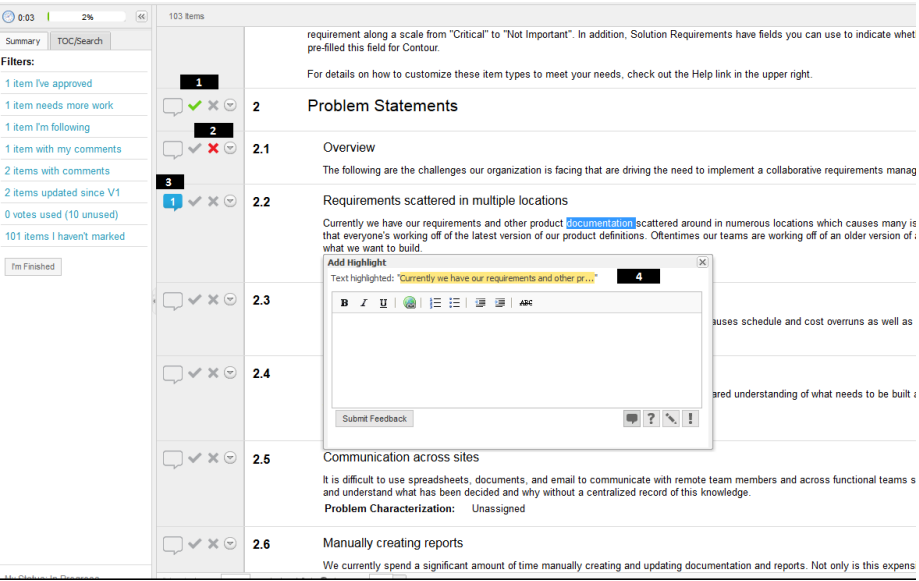


Figura 45: Opciones para los aprobadores de requisitos



El proceso de revisión finaliza una vez que el moderador considera que se han realizado las acciones necesarias por parte de los participantes en la revisión mediante un cierre manual o cuando el plazo fijado para su realización ha concluido.

Mediante esta utilidad Jama permite refinar la calidad en los requisitos hasta que son corregidos y aprobados en los procesos de revisión.

#### **2.3.5.2.2. Gestión de la calidad en Jama:**

Pese a no contar con ningún proceso de análisis semántico sobre los requisitos que ayude a determinar si se ha redactado con la calidad suficiente, Jama goza de otras características que ayudan notablemente a mejorar la calidad de cada uno de sus elementos.

Entre todas ellas destaca el trabajado proceso de revisión que incorpora, explicado en el apartado anterior, mediante el cual los requisitos son analizados y depurados por todos los integrantes de la revisión hasta que alcanzan la calidad necesaria para recibir la aprobación de los revisores.

Otro punto a favor de Jama es la conectividad que aporta entre los integrantes del proyecto ofreciendo multitud de canales de comunicación e informando, si así se desea, de cualquier cambio realizado a todas las partes interesadas.

Mediante la trazabilidad de sus requisitos Jama ofrece una utilidad de análisis de impacto para conocer que componentes pueden verse afectados por una modificación, ofreciendo así un gran control sobre la gestión y mantenimiento de todos los requisitos.

#### **2.3.5.2.3. Conclusiones sobre Jama:**

Jama es una herramienta con una curva de aprendizaje moderada facilitada por su manejo intuitivo y la gran cantidad de manuales de los que dispone.

Los múltiples tipos de elementos que ofrece para almacenar los componentes de un producto cubren las necesidades que puedan surgir en su desarrollo y la trazabilidad de cada cambio realizado en ellos permite ver la evolución de cada requisito desde el momento de su creación, no perdiendo ningún detalle en su evolución.

En cuanto a la preservación de la calidad en sus requisitos el punto fuerte de Jama es el proceso de revisiones, que facilita las actividades a realizar tanto por el moderador como por los revisores. El moderador puede ver en tiempo real qué requisitos han sido revisados, cuales son correctos, cuales son incorrectos y qué modificaciones que necesitan. Los revisores ven facilitado su trabajo mediante el método de marcaje de requisitos y la realización de comentarios y anotaciones para compartir información y opiniones con el resto de integrantes del proceso de revisión.

No obstante y al carecer de una utilidad de análisis semántico sobre la redacción de los requisitos todo el peso sobre el cuidado de la calidad del requisito recae sobre los desarrolladores, los cuales son en última instancia los que decidirán el nivel de calidad de cada uno pudiendo ser descuidados en este proceso.

Por otro lado, al tratarse de una herramienta en la nube, consta de los pros y contras de este tipo de herramientas. A su favor, que es accesible desde cualquier equipo con conexión a internet y toda la información está centralizada. En su contra, que tienes dependencia para su utilización tanto de la conexión en tu lugar de trabajo como la disponibilidad de los servidores de la herramienta.

Jama Software sigue trabajando y perfeccionando este producto incluyendo módulos nuevos según evolucionan las necesidades de los clientes. No obstante y como ya se comentó anteriormente deja todo el peso de la calidad de sus requisitos bajo el criterio de sus desarrolladores que pueden optar por realizar requisitos de mala calidad en beneficio de los tiempos de entrega como viene siendo habitual en muchos desarrollos.

### 2.3.5.3. Requirements Quality Analyzer - The Reuse Company:

*Requirements Quality Analyzer "RQA"* es una herramienta que pertenece a la suite de herramientas *Requirements Quality Suite "RQS"* desarrollada por The Reuse Company. Esta suite proporciona un conjunto de herramientas destinadas a configurar, administrar, analizar y mejorar la calidad de las especificaciones de requisitos aportando soluciones basadas en ontologías.

The Reuse Company es una marca perteneciente al Centro de Innovación y Soluciones Empresariales y Tecnológicas "CISSET" [26].

La suite *RQS* está compuesta por 3 herramientas:

- *Requirements Authoring Tool "RAT"*: RAT es una herramienta que ejerce como asistente a los autores de los requisitos para la redacción de sus especificaciones. Mediante el uso de unas plantillas previamente diseñadas denominadas "boiler plates", en las que se configuran las ontologías y normas gramaticales que se utilizarán en los requisitos, RAT ayuda en la redacción de los mismos permitiendo únicamente el uso de términos y estructuras incluidas en el "boiler plate". Mediante el uso de esta herramienta se asegura que los requisitos serán correctamente formados y entendibles tanto para las personas que los interpretan como para los procesos automáticos posteriores que los tengan que analizar.
- *Knowledge Manager "kM"*: kM es una herramienta para la gestión y mantenimiento de los tesauros y ontologías recopilados y utilizados por la empresa. Permite definir y gestionar los conceptos y semánticas de los proyectos así como sus relaciones. kM ejerce como repositorio de todo el conocimiento adquirido por la empresa el cual es reutilizable en cualquiera de sus desarrollos.
- *Requirements Quality Analyzer "RQA"*: RQA es una herramienta diseñada para evaluar la calidad de las especificaciones de requisitos. La aplicación permite la generación de una base de datos de métricas las cuales son aplicables a uno o varios requisitos para su evaluación. Mediante el uso de ontologías y técnicas semánticas RQA analizará la calidad de los requisitos incluidos en la especificación.

RQA es una aplicación de escritorio que actúa como la parte cliente de un servidor. La aplicación *Requirements Quality Suite Server* ejerce como servidor para la suite de



aplicaciones RQS y se encarga de gestionar los permisos de usuario, las licencias y almacenar las métricas y ontologías utilizadas por la compañía para que estén actualizadas y disponibles para todos los puestos que utilicen RQA centralizando así las bases de datos de conocimiento para que estén actualizadas en todo momento y para todos los usuarios.

RQA ha sido elegida para este trabajo por ser una herramienta centrada en el análisis semántico de los contenidos de los requisitos. Utilizándola junto a otras herramientas de gestión de requisitos resulta el complemento perfecto para asegurar la calidad de los mismos.

2.3.5.3.1. **Funcionalidades principales:**

RQA es una potente herramienta de análisis que además contiene multitud de características adicionales tales como la elaboración de gráficos y reportes a medida sobre la calidad de los requisitos o su conectividad directa con otras aplicaciones de gestión de requisitos como IBM Rational DOORS, Reqtify o Microsoft Excel pudiendo analizar directamente los requisitos almacenados en estas herramientas.

Las principales funcionalidades que ofrece RQA son:

2.3.5.3.1.1. **Aseguramiento de la calidad:**

Esta parte de la aplicación esta dedicada a la gestión de las métricas que serán utilizadas para medir y evaluar la calidad del contenido de los requisitos. De forma predefinida RQA proporciona 2 métricas “*INCOSE Guide Configuration*” conjunto de métricas proporcionado por INCOSE (International Council on Systems Engineering, site oficial en [27]) y “*TRC RQS Configuration*” conjunto de métricas proporcionado por The Reuse Company.

Todas las métricas son editables para adaptarse a las necesidades del desarrollo o pueden añadirse nuevas métricas si fuera necesario. El aspecto de esta ventana puede verse en la figura 46.

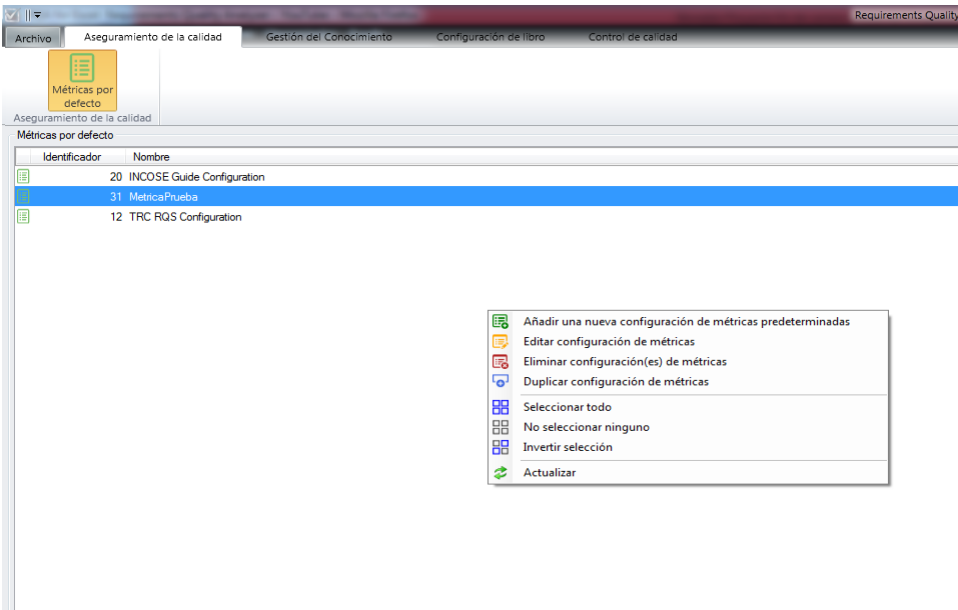


Figura 46: Configuración de métricas



Cada métrica tiene un conjunto de reglas que se aplicarán en el análisis de los requisitos para determinar su calidad. Estas reglas son configurables para adaptarse a las necesidades del desarrollo del producto, además permite la posibilidad de importar conjuntos de métricas de otros desarrollos. Como ejemplo, en la figura 47 se muestra la configuración de la regla de conteo de palabras la cual establece rangos de palabras para determinar su calidad. Calidad baja para rangos de 0 a 4 palabras o más de 61, calidad media para rangos de 36 a 61 palabras y calidad alta para requisitos con entre 4 y 36 palabras. Todos los rangos son editables y pueden añadirse más rangos si fuese necesario.

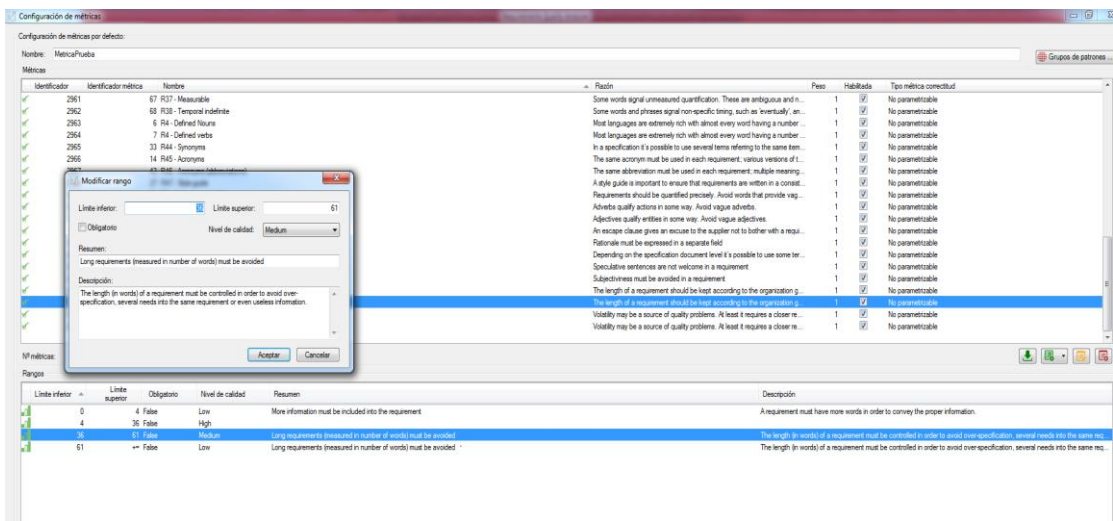


Figura 47: Configuración de reglas en las métricas

2.3.5.3.1.2. **Gestión del conocimiento:**

Ésta es la sección de la aplicación que se encarga del almacenamiento y gestión de los tesauros y ontologías que se aplicarán en los análisis de los requisitos.

La base de datos del conocimiento se separa en 4 partes: magnitudes, sentencias especiales, sustantivos de la ontología y verbos de la ontología.

2.3.5.3.1.2.1. **Gestión de magnitudes por defecto:**

Aquí se recoge el conocimiento de todas las unidades de medida aplicables en las definiciones de los requisitos. Mediante la definición y acotación del uso de las magnitudes se evitan descripciones erróneas o medidas que pueden ser no entendidas por los participantes del proyecto dando lugar a errores en su diseño y desarrollo. La forma de recoger este conocimiento parte de un elemento que puede ser medido (tiempo, longitud, peso, fuerza...), a continuación se determinan los sistemas métricos utilizados para medir dichos elementos y por último las unidades de medida permitidas dentro de estos sistemas métricos tal y como se muestra en la figura 48. Se pueden añadir o editar elementos de medida, así como los sistemas métricos para cuantificarles y las unidades de medida incluidas en dichos sistemas guardando las



modificaciones en la base de datos del servidor para ser transferidas a todos los integrantes del proyecto.

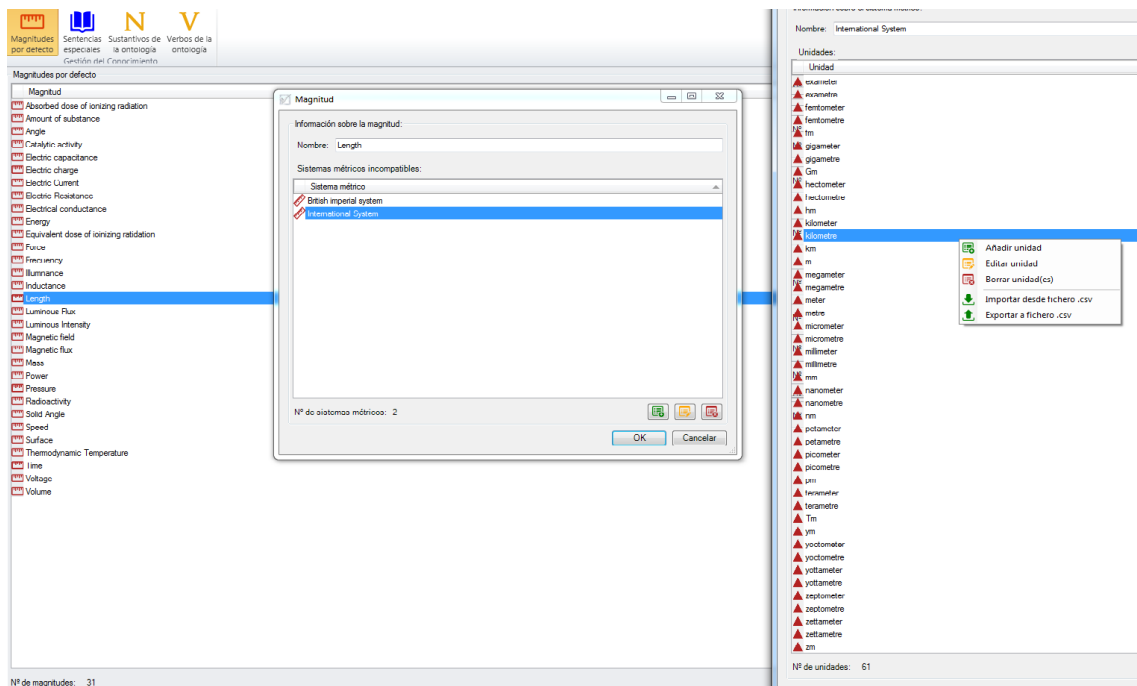


Figura 48: Gestión de magnitudes



2.3.5.3.1.2.2. Gestión de sentencias especiales:

En esta funcionalidad se gestiona el conocimiento sobre el uso y aplicación de ciertas sentencias que requieren un tratamiento particular dentro de la semántica de los requisitos. Términos ambiguos, sentencias subjetivas, sentencias especulativas, artículos indefinidos y demás términos se detectan y almacenan en este repositorio de conocimiento para ser aplicados en los análisis de los requisitos.

La herramienta permite editar, borrar y añadir nuevas sentencias o conjuntos de sentencias que serán detectadas durante los análisis.

En la figura 49 se muestra el repositorio por defecto para sentencias ambiguas.

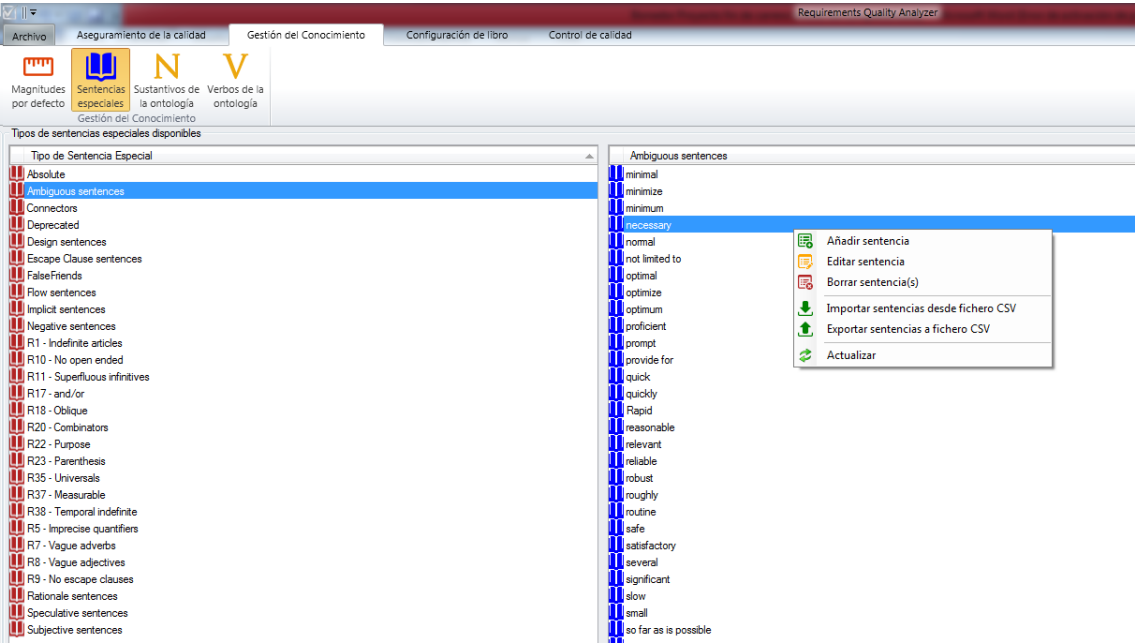


Figura 49: Configuración de sentencias especiales

2.3.5.3.1.2.3. Gestión de sustantivos de la ontología:

Para el análisis semántico de las oraciones incluidas en las descripciones de los requisitos RQA utiliza un repositorio de los sustantivos que se utilizarán en el desarrollo del producto el cual se gestiona desde esta funcionalidad.

Se pueden editar, borrar y añadir sustantivos a la base de datos de conocimiento para potenciar la robustez de los análisis.

En la figura 50 se muestra el contenido de esta pantalla así como la tipología de los sustantivos y sus cualidades definidas en la columna “cluster”.



Gestión del Conocimiento				
Archivo	Aseguramiento de la calidad	Gestión del Conocimiento	Configuración de libro	Control de calidad
Magnitudes por defecto	Sentencias especiales	Sustantivos de la ontología	Verbos de la ontología	
Término	Nota de alcance	Etiqueta sintáctica	Clúster	
.net		NOUN	< Sin «Clúster» >	
<null>		NOUN	< Sin «Clúster» >	
* c		MEASUREMENT UNIT	< Sin «Clúster» >	
Acceleration		NOUN	«Accelerate»	
Accelerometer		NOUN	«Accelerate»	
Action		NOUN	< Sin «Clúster» >	
Activation		NOUN	«Start»	
Actor		NOUN	«STAKEHOLDER»	
Addition		NOUN	«Add»	
Adds		NOUN	< Sin «Clúster» >	
Administrator		NOUN	< Sin «Clúster» >	
Adu		ACRONYMS	< Sin «Clúster» >	
Alert		NOUN	«Communication»	
Alignment		NOUN	«Align»	
Approval		NOUN	«Validate»	
Apu		ACRONYMS	< Sin «Clúster» >	
Assumption		NOUN	«Suppose»	
Automotion		NOUN	< Sin «Clúster» >	
Availability		NOUN	«AVAILABILITY : AGENT»	
Backup power		NOUN	«PROPERTY AND PHYSICAL CHARACTERISTIC»	
Beginning		NOUN	«Start»	
Buyer		NOUN	«Acquire»	
C		MEASUREMENT UNIT	< Sin «Clúster» >	
Cancellation			«Stop»	
Capability			< Sin «Clúster» >	
Car			< Sin «Clúster» >	
Celsius		MEASUREMENT UNIT	< Sin «Clúster» >	
Centigrade		MEASUREMENT UNIT	< Sin «Clúster» >	
Centimeter		MEASUREMENT UNIT	< Sin «Clúster» >	
Colour			«PROPERTY AND PHYSICAL CHARACTERISTIC»	
Commission		NOUN	«STAKEHOLDER»	
Compensation		NOUN	«Correct»	
Compliance		NOUN	«COMPLIANCE : ACTION»	
Component		NOUN	«SYSTEM ELEMENT»	
Concentration		NOUN	«Focus»	
Condition		NOUN	< Sin «Clúster» >	

Figura 50: Sustantivos de la ontología

2.3.5.3.1.2.4. Gestión de verbos de la ontología:

De igual forma que con los sustantivos, RQA proporciona una pantalla para la gestión de los verbos contenidos en la ontología para la realización de los análisis semánticos de los requisitos. Se puede editar, aumentar o borrar la cantidad de verbos incluidos en este repositorio.

En la figura 51 se muestra el contenido de esta pantalla con los distintos verbos incluidos así como sus tipos y sus cualidades indicadas en la columna “cluster”.

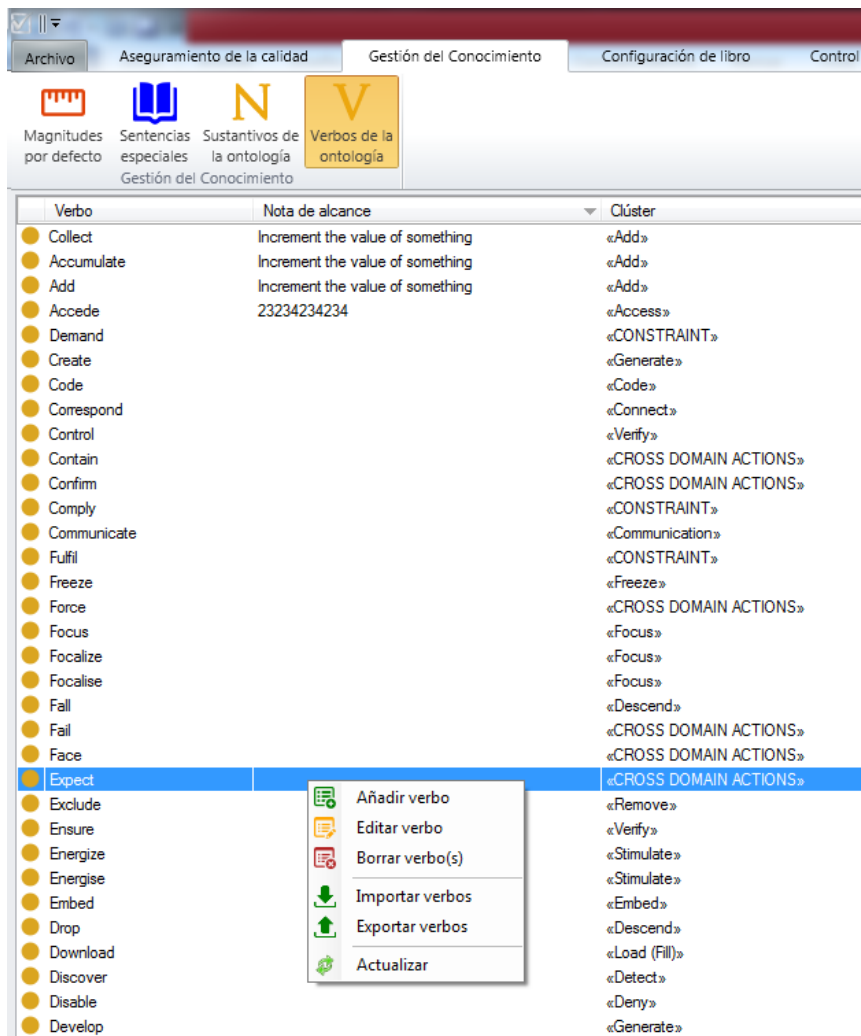


Figura 51: Verbos de la ontología

2.3.5.3.1.3. **Gestión del conocimiento:**

Mediante la pestaña de configuración de libro (para el caso de RQA con MS Excel) podemos asignar qué métricas y qué magnitudes van a ser aplicadas para los análisis de los requisitos contenidos en el libro Excel.

Las funciones fundamentales que realizan mediante esta pestaña son:

2.3.5.3.1.3.1. **Asignación de métricas a hojas:**

En esta pestaña se pueden asignar las distintas métricas definidas en la base de datos de conocimiento a cada una de las hojas que componen el libro Excel las cuales contienen conjuntos de requisitos.

RQA permite realizar análisis de calidad de forma manual aplicando las métricas asignadas a cada hoja o bien realizar análisis automáticos los cuales son programados para que se realicen de forma periódica. En la figura 52 se muestra la funcionalidad de esta pestaña.



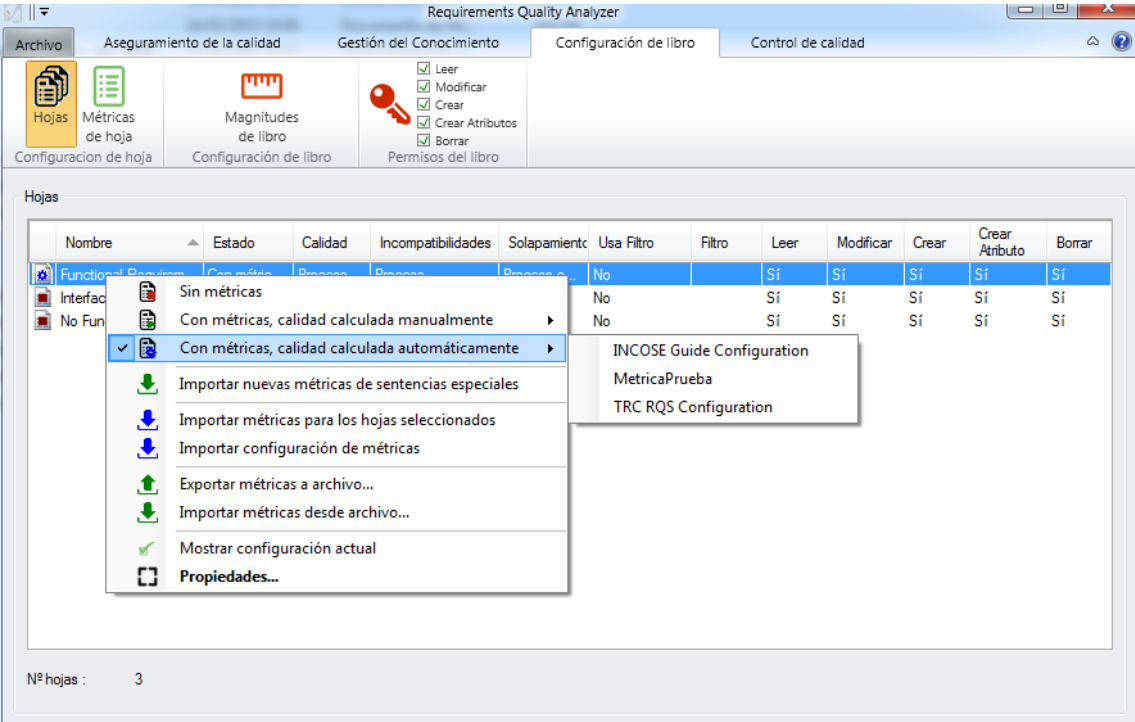


Figura 52: Asignación de métricas a hojas

2.3.5.3.1.3.2. Métricas de hojas:

En esta opción se pueden configurar las métricas que han sido previamente asignadas a cada hoja, permitiendo así una mayor personalización de los análisis para los distintos conjuntos de requisitos (separados en cada una de las hojas del libro) tal y como se muestra en la figura 53.

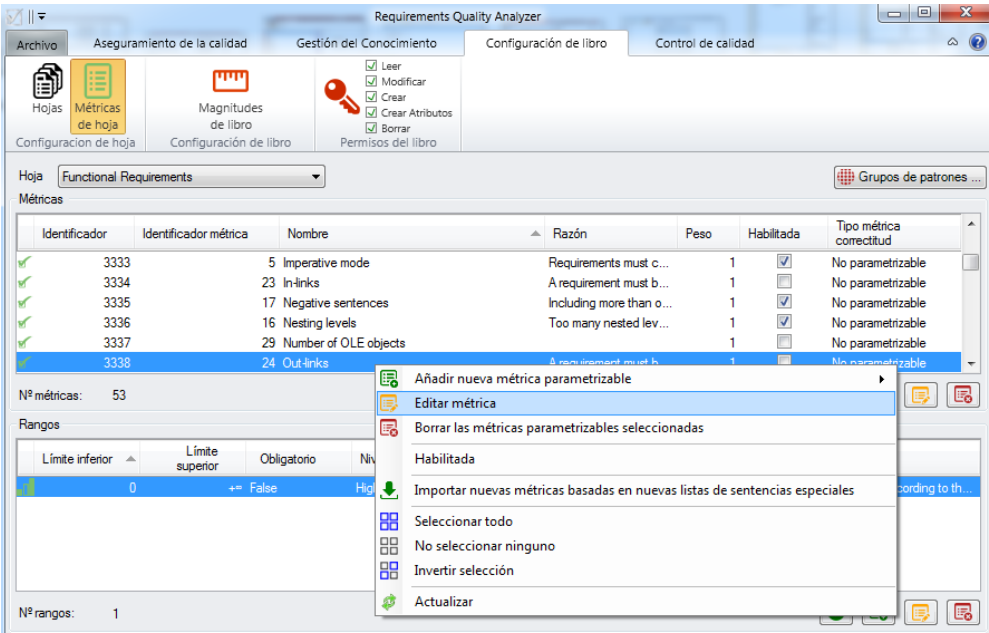


Figura 53: Configuración de métricas en cada hoja



Además en cada hoja se pueden definir los grupos de patrones a los que se ajustan los requisitos almacenados en cada hoja a fin de facilitar su análisis. Cada patrón contiene una serie de reglas que son características de cada tipo de requisito. La asignación de patrones se muestra en la figura 54.

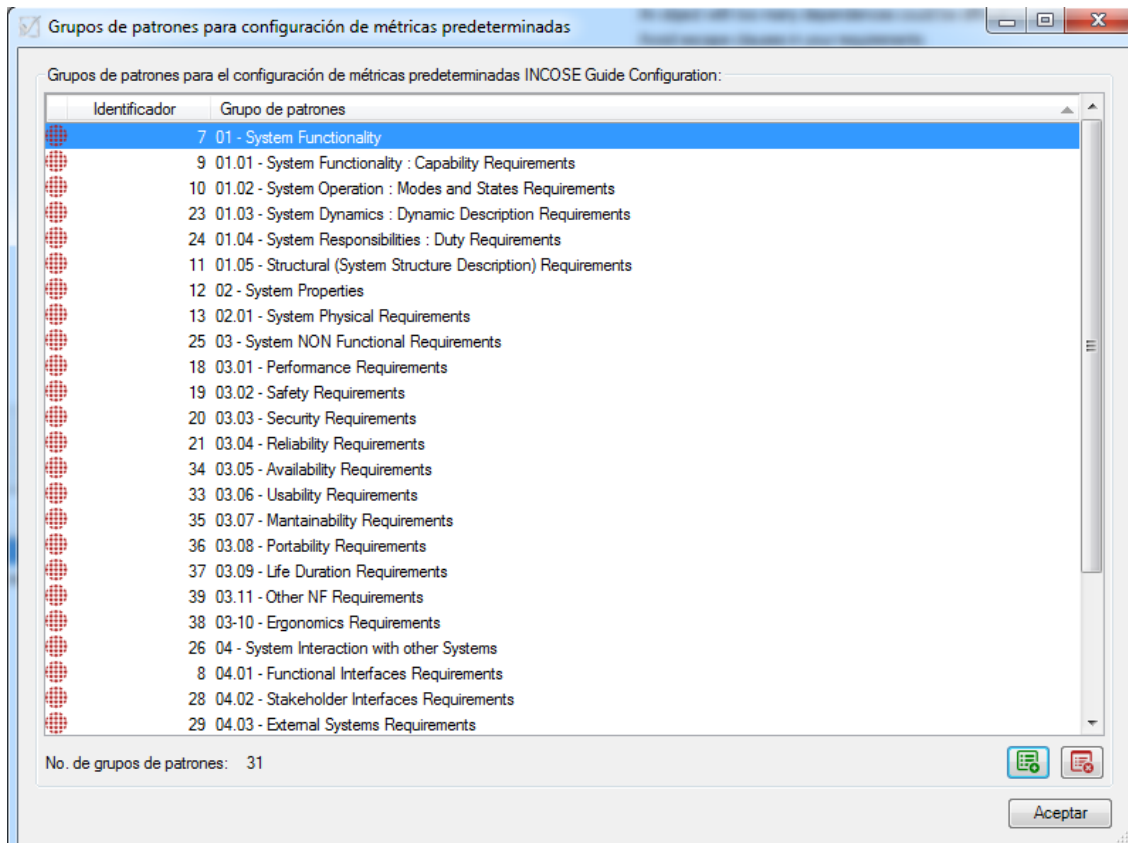


Figura 54: Grupos de patrones

2.3.5.3.1.3.3. Magnitudes de libro:

De igual forma que con las métricas a cada libro se le asigna un conjunto de las magnitudes contenidas en la ontología los cuales utilizará para realizar el análisis semántico de cada requisito a fin de determinar su calidad. Como se muestra en la figura 55 se pueden añadir, eliminar o editar los conjuntos de magnitudes asignadas al libro.

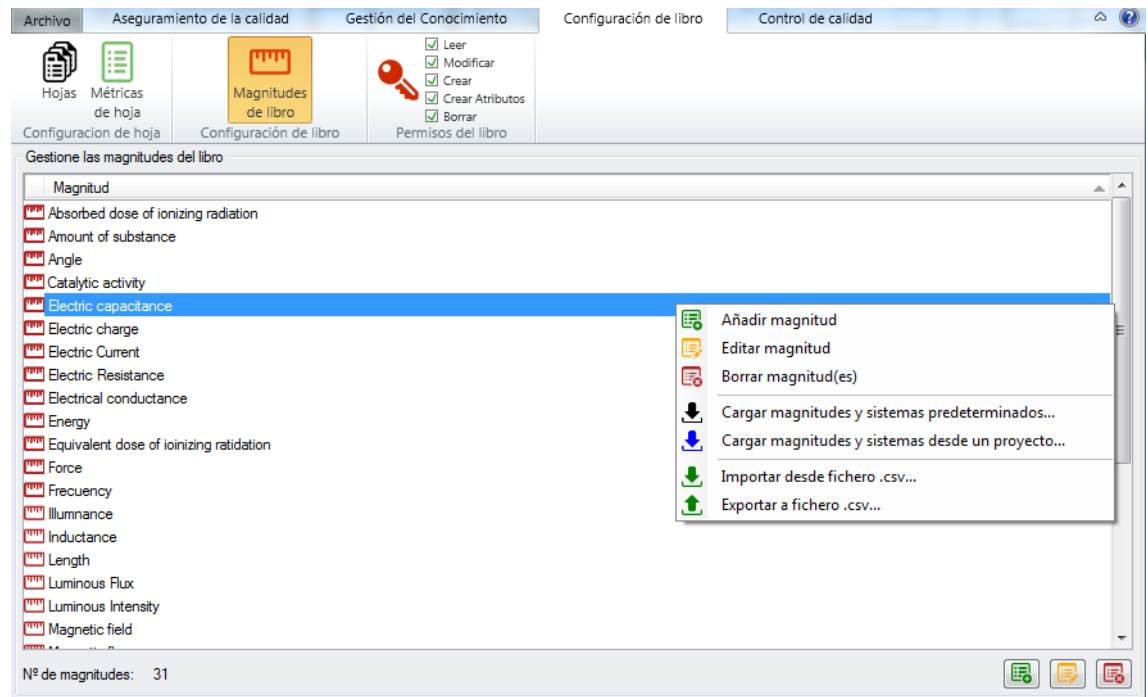


Figura 55: Configuración de magnitudes del libro

#### 2.3.5.3.1.4. Control de calidad:

En las funcionalidades contenidas en la pestaña de “Control de calidad” se muestran los resultados de los procesos de control de calidad tras la realización de los análisis aplicando las métricas previamente configuradas en cada una de las especificaciones de requisitos.

Las acciones que se pueden realizar desde esta pestaña se exponen a continuación:

##### 2.3.5.3.1.4.1. Control de calidad – Cuadro de mando:

El cuadro de mando es una de las principales funcionalidades de RQA. Mediante el cuadro de mando se realiza el cálculo manual de la calidad de los requisitos contenidos dentro de la especificación mediante las métricas que les han sido previamente asignadas. En esta vista se muestra el identificador del requisito, la descripción, su módulo padre, el nivel de calidad del requisito (bajo = 1 estrella, medio = 2 estrellas, alto = 3 estrellas), una puntuación sobre el peso del requisito y la fecha en la que ha sido calculada la calidad. En la figura 56 se muestra el contenido inicial de esta vista.



Requisitos				
Calidad según: < Todas las métricas >				
Identificador	Texto	Módulo padre	Calidad	Puntuación / Fecha de calidad
REQ001	Every painting must always belong to only one Theme.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:24
REQ002	The system has a drop-down list of Themes already created for an administrator to choose from while adding a new painting information.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:24
REQ003	The graphical user interface (GUI) should allow the information of only one painting to be entered at a time.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:24
REQ004	The GUI should allow the administrator to create new painting information, review and edit existing painting information and save them. If painting information section with the same title already exists, the GUI should prompt administrator accordingly.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:25
REQ005	The Virtual/Ver application should allow the administrator to save any information about any painting, artist, gallery and location on the "Revised, Inc." server at any point during editing.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:25
REQ006	An administrator should be able to perform all the search queries as a normal user. Moreover, he should be able to edit any of the information sections (like paintings, artists, themes, galleries, countries) and is responsible for updating and maintaining information links.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:25
REQ007	He should be able to delete any existing information section. Before deleting an existing information section, the application should elicit a confirmation from him. When any section is deleted, all the links that point to it are broken. Moreover, only one information section can be de.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:25
REQ008	The user should be able to view any information section, the results of any kind of search and a list of files of paintings that have been put in the user's Favorites folder.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:25
REQ009	Any entity/text on the user interface that is a link should be in blue font and underlined.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:25
REQ010	The information in any "Painting Information" should have the title of the painting on top, followed by the image, the artist, the date the painting was created, the theme of the painting, the name and location of the art gallery where the painting is located and the description of the.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:26
REQ011	The information in any "Artist Information" must have the name of the artist, the list of paintings he created and the description of the artist.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:26
REQ012	The information in any "Theme Information" should have the name of the theme, the list of paintings that fall under that theme and the description of the theme.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:26
REQ013	The information in any "Gallery Information" should have the name of the gallery, the location of the gallery, the list of paintings in the gallery and the description of the gallery.	Functional Require.	★ ★ ★	20.00 11/06/2015 20:27:26
REQ014	The program (Access) uses the hard disk. Access to the hard drive and other hardware is managed by the operating system and Access.	Interface Requirem.	★ ★ ★	2.87 11/06/2015 20:27:29
REQ015	If we decide to implement an Ad Hoc network for a shared database, the operating system will handle those connections.	Interface Requirem.	★ ★ ★	1.70 11/06/2015 20:27:29
REQ016	The Access system may be used to import and export data with Microsoft Excel. This functionality is built in to the user interface.	Interface Requirem.	★ ★ ★	2.87 11/06/2015 20:27:29
REQ017	The user interface for this program is the interface provided by Microsoft Access 2007. Access includes forms and reports for the users to query and organize data to suit their needs. Forms and reports both have builders that let the user specify which fields they want to use and.	Interface Requirem.	★ ★ ★	3.19 11/06/2015 20:27:30
REQ018	The Background Task Manager shall display status messages in a designated area of the user interface at intervals of 60 plus or minus 10 seconds.	Interface Requirem.	★ ★ ★	1.70 11/06/2015 20:27:30
REQ019	Background task processing is progressing normally, the percentage of the background task processing that has been completed shall be displayed.	Interface Requirem.	★ ★ ★	1.91 11/06/2015 20:27:30
REQ020	An error message shall be displayed if the background task has stalled.	Interface Requirem.	★ ★ ★	1.17 11/06/2015 20:27:30
REQ021	The system shall be designed with a level of security appropriate for the sensitivity of information enclosed in the database. More interaction is needed with client about the validity of the information. Since there is no obvious information that is of a high security level such as or.	No Functional Req.	★ ★ ★	3.93 11/06/2015 20:27:33
REQ022	This system will be compatible with any computer that has Microsoft Office Professional 2007 or later installed (whether PC or Mac), and will be designed with more than one computer in mind.	No Functional Req.	★ ★ ★	3.72 11/06/2015 20:27:34
REQ023	Reliability is one of the key attributes of the system. Back-ups will be made regularly so that restoration with minimal data loss is possible in the event of unforeseen events. The system will also be thoroughly tested by all team members to ensure reliability.	No Functional Req.	★ ★ ★	3.08 11/06/2015 20:27:34
REQ024	The system shall be maintained by Shalea Ross, of the ISA, or delegated to another employee.	No Functional Req.	★ ★ ★	20.00 11/06/2015 20:27:34
REQ025	The system shall be designed in a way that shall allow it to be run on multiple computers with Microsoft Office Professional 2007 or later installed.	No Functional Req.	★ ★ ★	3.29 11/06/2015 20:27:34
REQ026	The system shall be designed and documented in such a way that anybody with an understanding of Microsoft Access shall be able to extend the system to fit their needs with the team's basic instructions.	No Functional Req.	★ ★ ★	3.08 11/06/2015 20:27:34
REQ027	The system should be designed in a way that allows the database to be re-used regularly for the various client auctions that the organization shall hold.	No Functional Req.	★ ★ ★	2.65 11/06/2015 20:27:34
REQ028	95% of transactions on the public-facing website portal shall be processed in less than 4s	No Functional Req.	★ ★ ★	20.00 11/06/2015 20:27:35

Figura 56: Control Calidad - Cuadro de mando

Mediante las opciones disponibles dentro del cuadro de mando se encuentran:

- Creación de informes: RQA permite la creación de informes individuales o de conjuntos de requisitos para su explotación.
- Recalculo de la calidad: Si un requisito ha sido editado mediante la opción de “Recalcular calidad” se volverá a realizar el análisis del requisito para aplicar las métricas sobre el nuevo contenido.
- Ver Calidad: Pulsando el botón de “Ver Calidad” RQA muestra el detalle de los análisis semánticos aplicados por cada métrica en el requisito seleccionado.

Mediante la visualización detallada del análisis el usuario puede realizar las correcciones adecuadas en base a la información mostrada por RQA cuyos elementos incorrectos resalta en color rojo como puede observarse en la figura 57 mediante el detalle de la métrica de “conceptos fuera de dominio” o en la figura 58 mediante el detalle de la métrica de “uso de voz pasiva”.

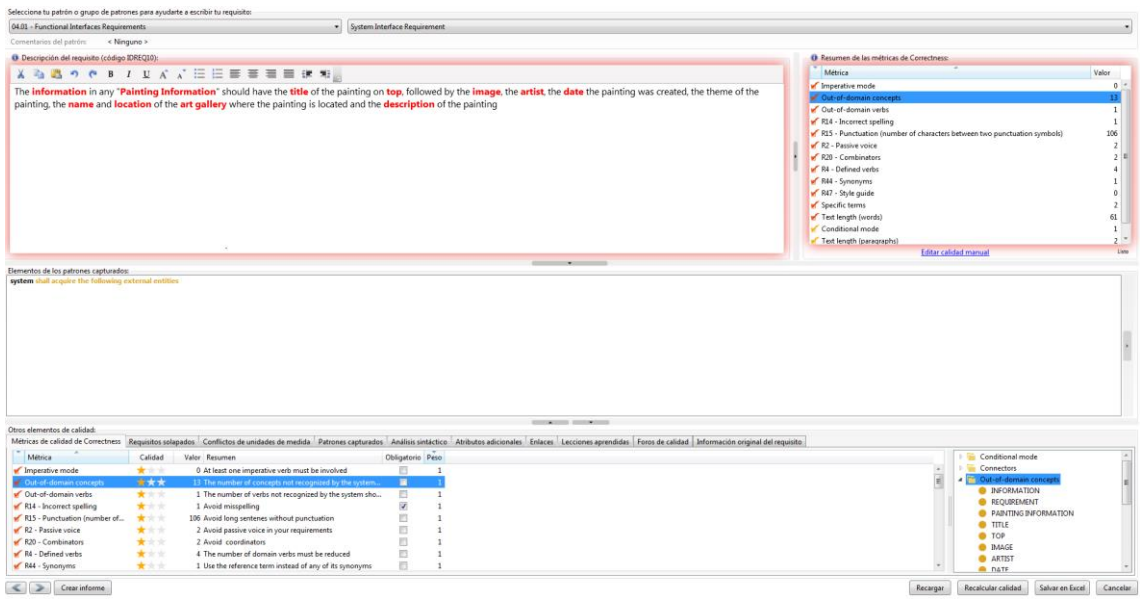


Figura 57: Vista de Calidad: Conceptos fuera de dominio

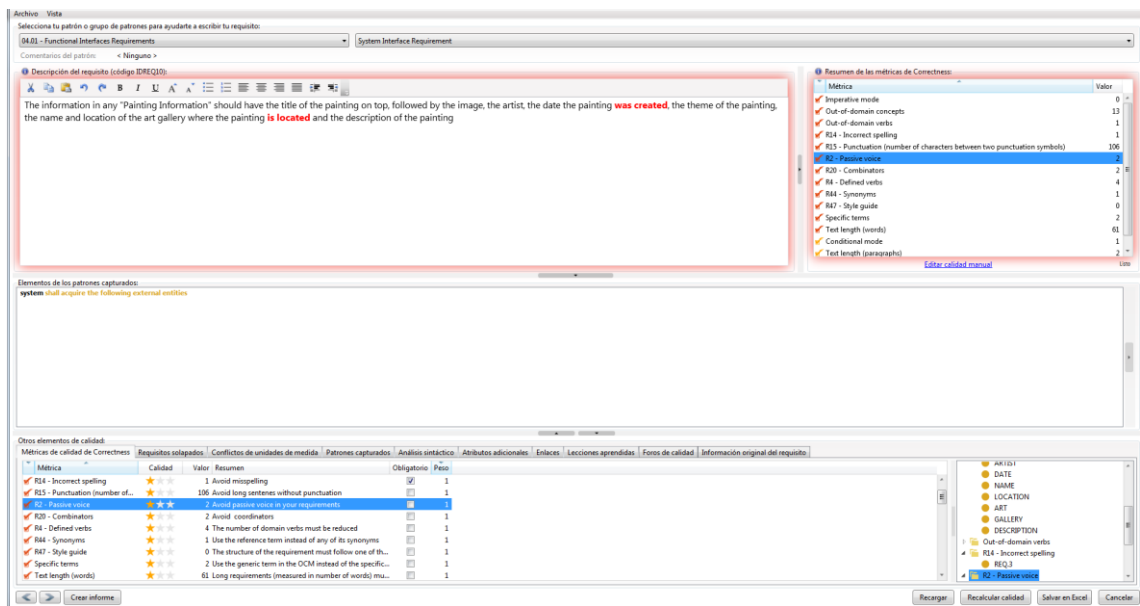


Figura 58: Vista de Calidad: Uso de voz pasiva

Cada una de las métricas aplicadas sobre los requisitos muestra en su detalle recomendaciones para que el usuario realice las correcciones necesarias en la redacción del requisito. En la figura 59 se muestra un ejemplo sobre lo que la métrica recomienda para su corrección.

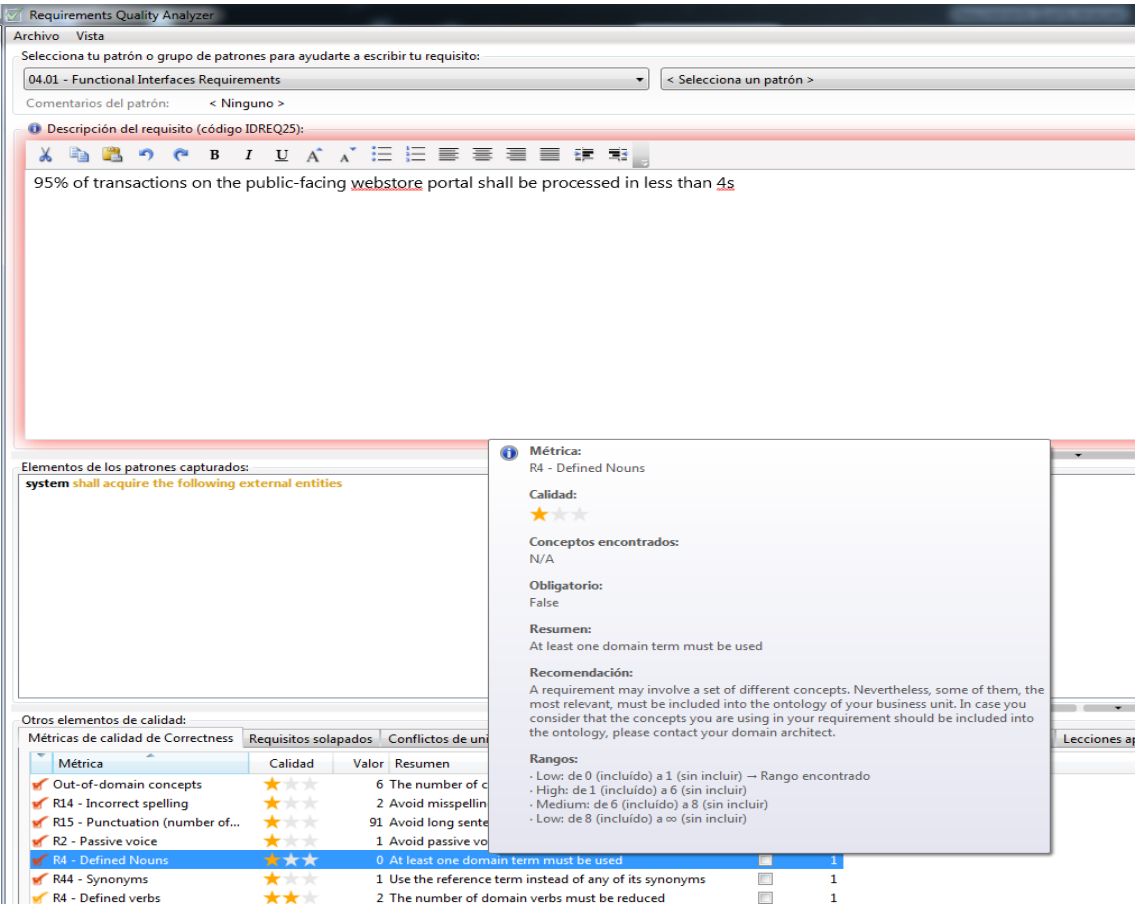


Figura 59: Vista de Calidad - Detalle recomendación métricas

2.3.5.3.1.4.2. Control de calidad - Métricas:

En este apartado se muestra el resultado de la aplicación de las métricas que han sido asignadas a las especificaciones de requisitos mostrando un resumen de los resultados aportados por cada métrica.

Entre los datos aportados se puede observar el conteo de requisitos que la cumplen tanto de forma satisfactoria como de forma insatisfactoria, los usuarios que mejor y peor han redactado los requisitos en base a una métrica concreta y una serie de estadísticas sobre su aplicación. En la figura 60 se muestra la información aportada en la opción de “métricas”.

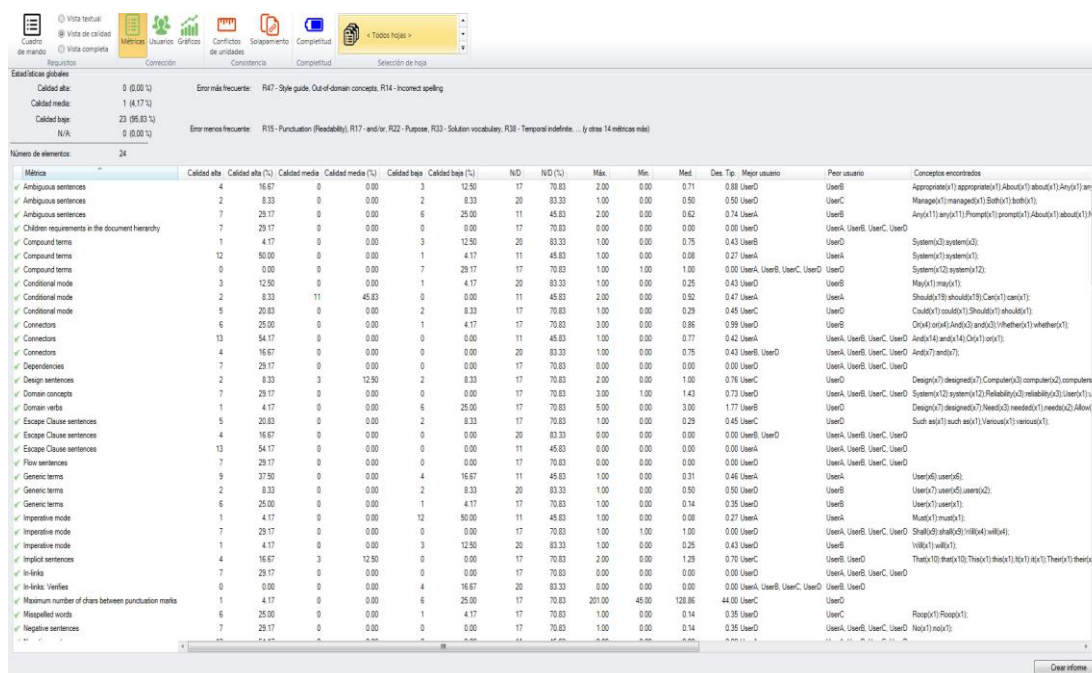


Figura 60: Control de calidad: Métricas

### 2.3.5.3.1.4.3. Control de calidad - Usuarios:

Mediante la vista de “Usuarios” en Control de Calidad RQA nos ofrece una serie de datos estadísticos sobre el trabajo realizado por cada usuario. Entre los datos que aporta se observa el número de requisitos para cada nivel de calidad que ha realizado el usuario mostrando un porcentaje sobre el total de sus requisitos, así como los errores que ha cometido con mayor frecuencia.

En el despliegue detallado se encuentra un resumen de cada métrica similar al mostrado en Control de calidad: Métricas.

En la figura 61 se muestra un ejemplo con los datos aportados por esta funcionalidad:

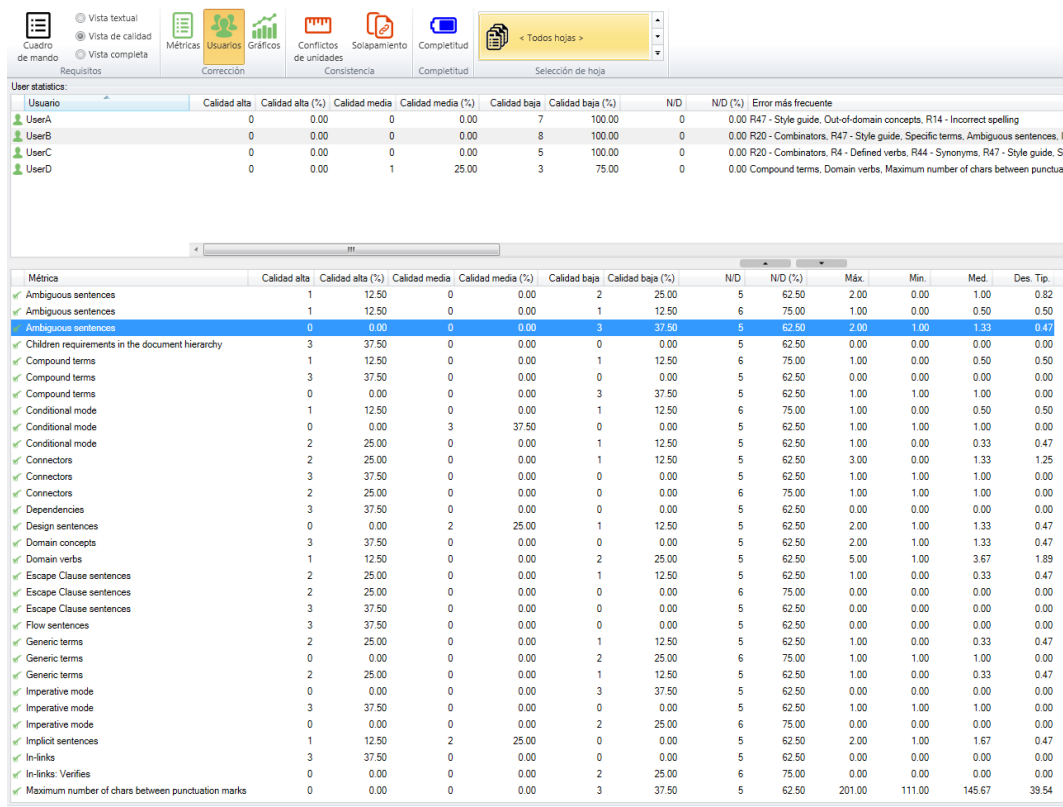


Figura 61: Control de calidad: Usuarios

2.3.5.3.1.4.4. Control de calidad – Conflictos de Unidades:  
RQA aporta esta funcionalidad para detectar automáticamente conflictos entre las unidades de medida aplicadas en los requisitos, asegurando así que en todo el conjunto de la especificación de requisitos se apliquen las mismas unidades de medida. En la figura 62 se muestra un ejemplo con la información aportada por esta funcionalidad, en la cual se observa la aplicación de 2 sistemas métricos distintos en la misma especificación.

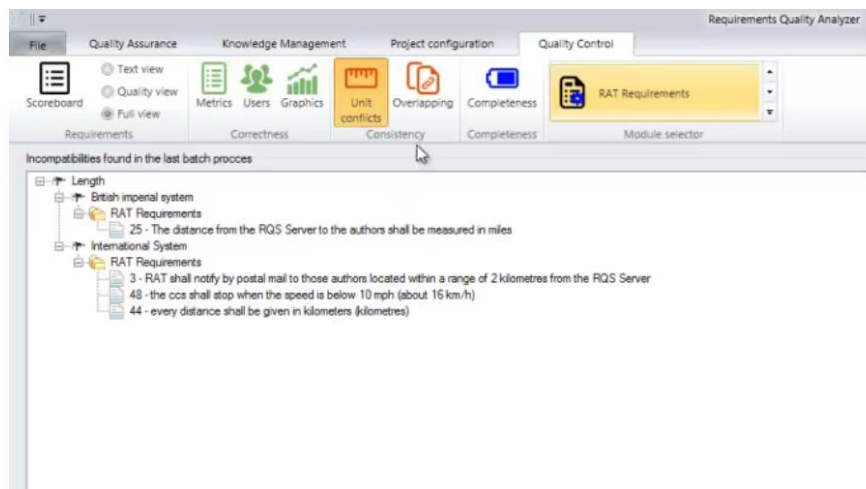


Figura 62: Control de calidad - Conflictos de Unidades





2.3.5.3.1.4.5. Control de calidad – Solapamiento:

RQA realiza un control sobre posibles solapamientos entre requisitos, es decir, que dos o más requisitos describan la misma funcionalidad con alguna diferencia. Mediante un análisis gramatical del contenido del requisito RQA detecta estructuras gramaticales similares en otros requisitos y los muestra en la ventana de solapamientos para que el usuario decida eliminar, corregir o mantener los requisitos con estas similitudes. En la figura 63 se muestra la información mostrada en el control de calidad de solapamientos.

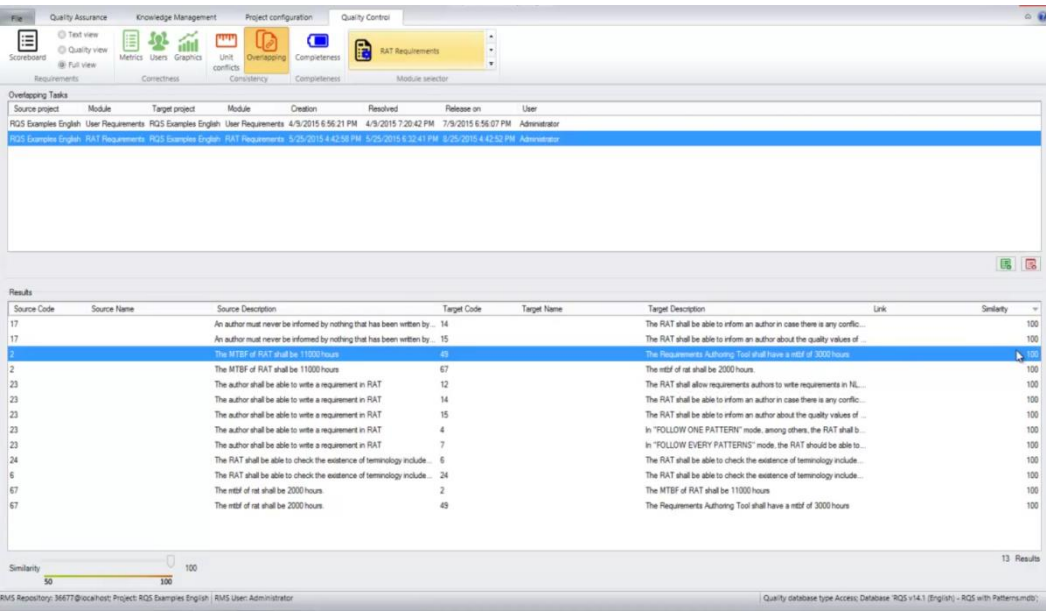


Figura 63: Control de calidad - Solapamientos

2.3.5.3.1.4.6. Control de calidad – Completitud:

La vista de completitud de RQA muestra el número de requisitos de cada tipo incluidos dentro de la especificación de requisitos, facilitando así una visión del conjunto total de requisitos redactados. En la figura 64 se incluye un ejemplo con la información que aporta esta funcionalidad de RQA.

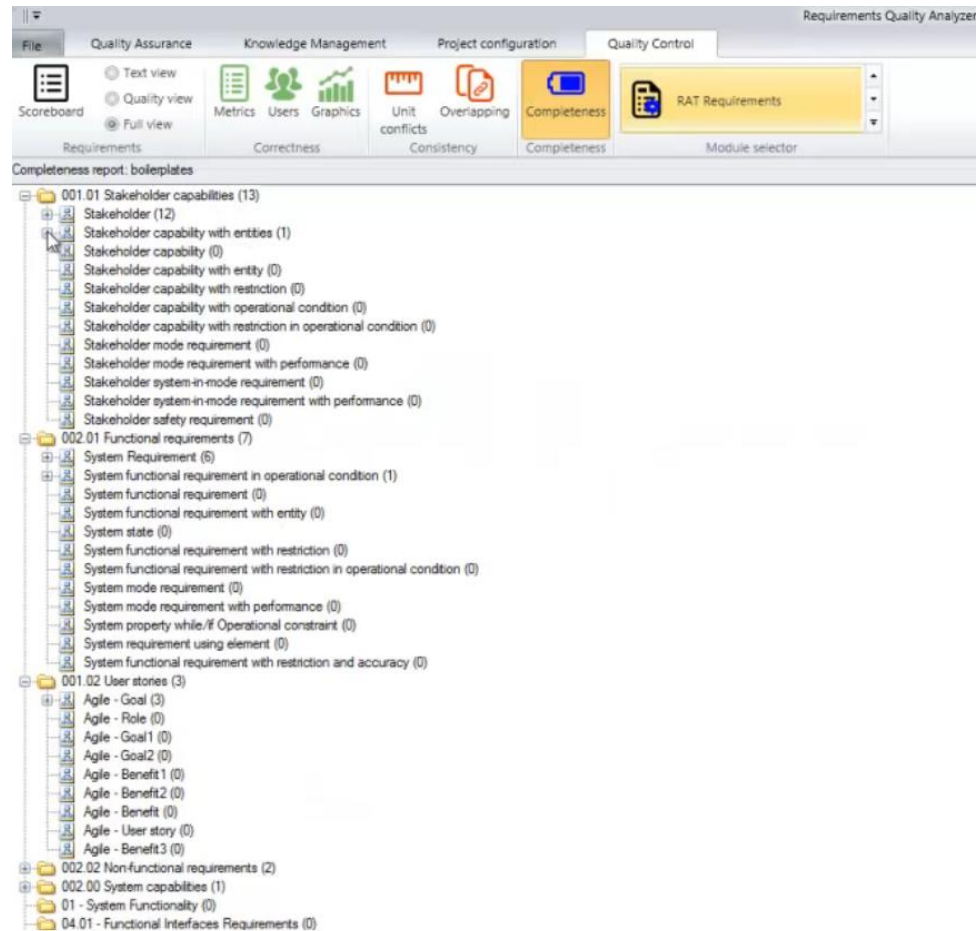


Figura 64: Control de calidad - Completitud

#### 2.3.5.3.2. Conclusiones sobre RQA:

RQA es una herramienta de análisis muy potente cuya utilización garantiza un buen nivel de calidad en la realización de los requisitos.

Su curva de aprendizaje es moderada, no obstante The Reuse Company facilita todo tipo de manuales y video tutoriales para que los usuarios se empiecen a familiarizar con su manejo.

Al tratarse de una herramienta que basa sus análisis en una serie de métricas apoyadas en una base de conocimiento formada por magnitudes, sentencias, verbos y sustantivos es necesaria una inversión de tiempo inicial en la configuración de las métricas y el enriquecimiento de la base de datos de conocimiento cuando se comienza a utilizar. Por otro lado, con el paso del tiempo y el desarrollo de diversos proyectos la potencia de la herramienta se va incrementando al contener cada vez más conocimientos dentro de sus ontologías con lo que el tiempo dedicado a la configuración de conocimientos y métricas se irá reduciendo paulatinamente hasta unos mínimos.

RQA es una herramienta de escritorio la cual se integra con otras herramientas de gestión de requisitos para realizar el análisis sobre la información almacenada por estas últimas. Actualmente RQA es compatible con *IBM DOORS*, *Reqtify*, *Visure Requirements* y *MS Excel*.

Al utilizar RQA junto a las otras herramientas de la suite RQS se aumenta de gran forma la capacidad de gestionar y garantizar la calidad de los requisitos. kM permite almacenar y centralizar todo el conocimiento adquirido en el desarrollo de los diversos proyectos de la compañía, pudiendo aplicar este conjunto de conocimiento a los nuevos proyectos. Mediante el uso de RAT para la redacción de los requisitos, nos aseguramos que las sentencias gramaticales de los requisitos están utilizando los términos disponibles en la base de datos de conocimiento (gestionada por kM) así como una estructura sintáctica sencilla y concisa que permitirá a RQA realizar un análisis mucho más fiable y profundo de cada requisito, garantizando así altos niveles de calidad en cada uno de ellos y minimizando en gran medida impactos negativos en el desarrollo de los proyectos.

La suite RQS y toda la información necesaria sobre cada una de sus herramientas se encuentran en [25].

### **3. Capítulo 3: Parte práctica – Cuestionario sobre calidad de requisitos:**

#### **3.1. Introducción:**

Para la realización de la parte práctica de este proyecto se ha elaborado un cuestionario dirigido a varias empresas con la finalidad de visualizar qué procesos, metodologías y tecnológicas utilizan para gestionar y mejorar la calidad de sus requisitos.

La encuesta se ha lanzado a empresas de distintos sectores y países a fin de ilustrar en sus resultados como influye el tipo de negocio realizado en el control de la calidad de los requisitos.

El cuestionario ha sido realizado utilizando la herramienta de “*Google Forms*” para la realización de formularios disponible vía web en [29].

#### **3.2. Estructura del cuestionario:**

El cuestionario se divide en 5 secciones:

##### **3.2.1. Características de la empresa:**

Preguntas sobre las características de la empresa encuestada. Se recoge información sobre su antigüedad, número de empleados, sectores de negocio, compendios de buenas prácticas utilizados y que normativas o estándares aplican de forma común en sus procesos de desarrollo.

**Figura 65: Cuestionario - Características de la empresa**

Recoge la información sobre el uso de ciertos procedimientos y técnicas de captura y almacenamiento de requisitos que permiten y asegura la calidad de los mismos. Estas técnicas han sido identificadas en las normativas y metodologías de buenas prácticas expuestas en este documento (ITIL, CMMI, ISO/IEC/IEEE 29148 – 2011, ISO/IEC/IEEE 12207:2008, ISO/IEC/IEEE 15288:2008...etc.).

**Figura 66: Cuestionario - Captura de requisitos**

### 3.2.3. Verificación y validación de requisitos:

En este apartado se recopila información sobre la aplicación de ciertos procedimientos de verificación y validación de requisitos destinados a mejorar la calidad de los mismos y depurar e identificar errores en los mismos.

Verificación y validación de los requisitos

36. ¿Se identifican y documentan los criterios de aceptación de cada requisito?

☐ Siempre  
☐ A veces  
☐ Nunca

37. ¿Se realiza una fase de revisión de requisitos por parte de un equipo de revisores antes de ser incluidos en la línea base?

☐ Siempre  
☐ A veces  
☐ Nunca

38. ¿Qué roles intervienen en los procesos de revisión de requisitos?  
 Marque los roles que participan

☐ Autor  
☐ Moderador  
☐ Lector  
☐ Registrador  
☐ Revisor  
☐ Otro:

39. ¿Se documentan los defectos detectados en la fase de revisiones?

☐ Siempre  
☐ A veces  
☐ Nunca

40. ¿Se analiza el impacto que suponen las soluciones propuestas a los defectos?

☐ Siempre  
☐ A veces  
☐ Nunca

Figura 67: Cuestionario - Verificación y validación

### 3.2.4. Gestión de los requisitos:

Se recoge información sobre ciertos procedimientos aplicables a la gestión de los requisitos como la gestión de las modificaciones o la gestión de las relaciones entre requisitos.

50. Todas las modificaciones ¿son documentadas independientemente de su impacto?

☐ Siempre  
☐ A veces  
☐ Nunca

51. ¿Se establecen canales de comunicación para transmitir las modificaciones realizadas a los requisitos a todas sus personas implicadas?

☐ Siempre  
☐ A veces  
☐ Nunca

52. ¿Se establece un plan de validación para las modificaciones sobre los requisitos?

☐ Siempre  
☐ A veces  
☐ Nunca

53. ¿Se muestran y actualizan los estados en los que se encuentra cada requisito? (Pendiente, en curso, realizado...etc)

☐ Siempre  
☐ A veces  
☐ Nunca

54. Cuando se modifica un requisito. ¿Se realizan estudios sobre el impacto que dicho cambio tiene sobre sus requisitos relacionados?


☐ Siempre  
☐ A veces  
☐ Nunca

Figura 68: Cuestionario - Gestión de requisitos



3.2.5. **Herramientas para requisitos:**

En esta sección se pregunta sobre las herramientas que son utilizadas para la captura, gestión y validación de los requisitos.



Autoevaluación sobre gestión de la calidad en los requisitos.

Herramientas para requisitos

56. ¿Se utiliza alguna herramienta para el almacenamiento y gestión de los requisitos?  
IBM Rational Doors, HP Quality Center, Jama ...

☐ Siempre  
☐ A veces  
☐ Nunca

57. En caso positivo, ¿cuál es la herramienta utilizada para este fin?

58. ¿Se utiliza alguna herramienta para el control de la trazabilidad de los requisitos?  
IBM Rational Doors, HP Quality Center, Jama ...

☐ Siempre  
☐ A veces  
☐ Nunca

59. En caso positivo, ¿cuál es la herramienta utilizada para este fin?

Figura 69: Cuestionario - Herramientas para requisitos



### 3.3. Exposición de los resultados del cuestionario:

En la realización de la encuesta han participado alrededor de 20 empresas de productos informáticos de diferentes países.

Se han escogido empresas con tamaños y objetivos distintos a fin de ilustrar de modo general el estado de la gestión de la calidad de los requisitos en las empresas actuales sin sesgar la participación a un tipo de empresa concreto.

A continuación se exponen los resultados más significativos de cada sección del cuestionario:

#### 3.3.1. Datos sobre características de la empresa:

Los datos obtenidos sobre las características de las empresas encuestadas muestran que la mitad tienen entre 1000 y 5000 empleados en su plantilla y son empresas que llevan en funcionamiento más de 10 años en su mayoría como puede apreciarse en la figura 70. Los sectores de mercado en los que trabajan se reparten como muestra la figura 71.

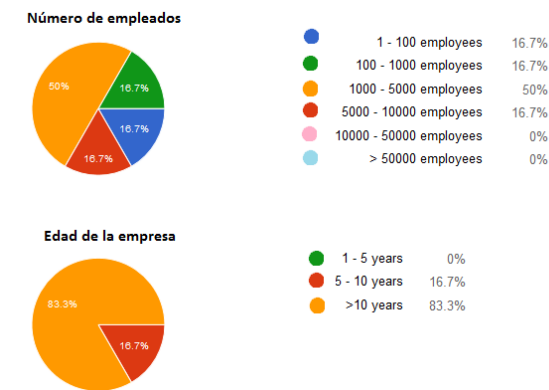


Figura 70: Empleados y edad de las empresas

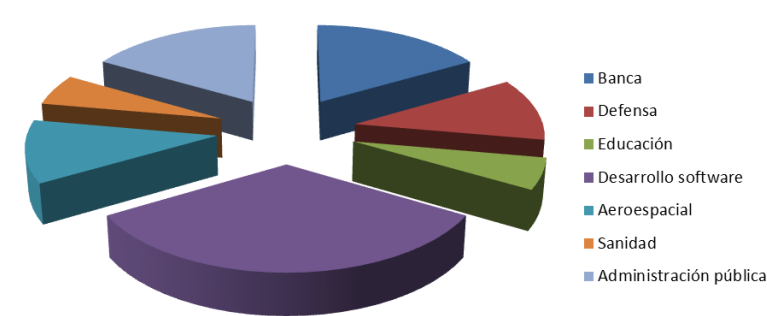


Figura 71: Sectores de mercado en los que participan



Los datos obtenidos sobre la utilización de compendios de buenas prácticas y normativas específicas en todos sus procesos podemos concluir que un 80% de los encuestados utilizan algún compendio de buenas prácticas en sus desarrollos siendo ITIL el compendio más común.

Un 20% tienen integrado el modelo de madurez CMMI en los desarrollos de sus proyectos hasta el grado 3 del mismo.

En cuanto a la utilización de normativas comunes al desarrollo de los proyectos (normativas aplicadas en todos los proyectos desarrollados por la empresa y no de forma específica para un proyecto puntual) observamos que son las empresas dedicadas a sectores como sanidad, defensa o aeroespacial las que hacen uso de este tipo de normativas, siendo las más comunes la ISO 9000 y la ISO 20000.

### **3.3.2. Datos sobre el proceso de captura de requisitos:**

Se ha preguntado a las empresas encuestadas acerca de los siguientes puntos:

- Si hacían uso de un diccionario de términos y acrónimos para usar en sus requisitos.
- Si utilizaban plantillas para recoger las especificaciones de requisitos.
- Si fijaban relaciones entre los requisitos estableciendo así jerarquías de dependencia entre ellos.
- Si fijaban prioridades en la realización de unos requisitos frente a otros.
- Si creaban escenarios para facilitar el entendimiento de los requisitos.
- Si detectaban y documentaban los riesgos derivados de no realizar un requisito.
- Si realizaban algún proceso para detectar requisitos contradictorios entre sí.
- Si hacían un uso delimitado del lenguaje en la realización de sus requisitos acotando el uso de verbos y formas verbales, sustantivos, adjetivo y demás elementos del lenguaje a un conjunto predeterminado a fin de facilitar el entendimiento y análisis de los mismos.

En la siguiente figura se observa el porcentaje de aplicación de estos métodos por las empresas encuestadas.

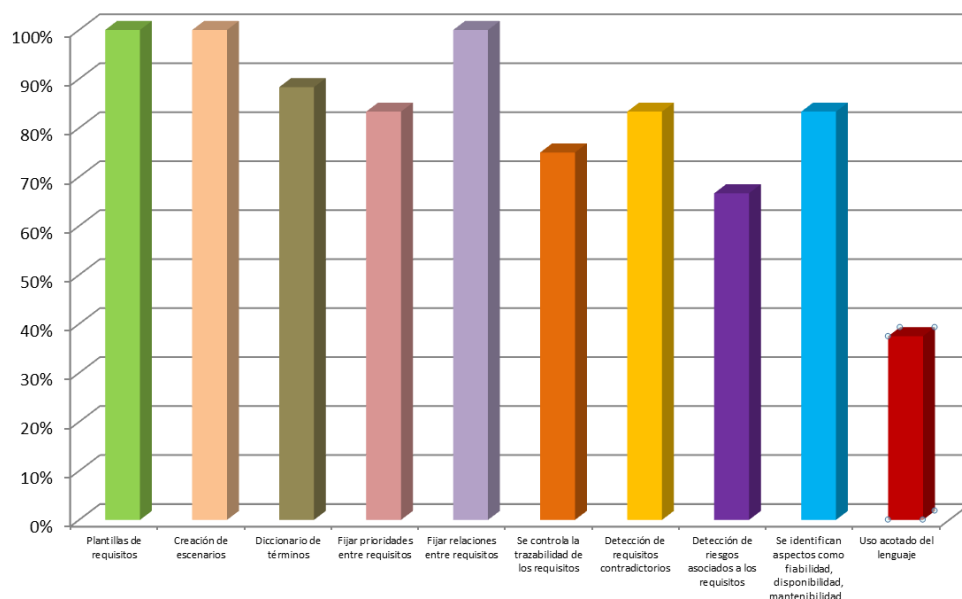


Figura 72: Métodos utilizados para la elaboración de requisitos

También se ha preguntado a los encuestados sobre el uso de la reutilización de los requisitos entre sus proyectos. De los resultados obtenidos observamos que el 66,7% de las empresas reutilizan requisitos que ya han desarrollado en casi todos sus proyectos mientras que un 16,7% solo lo hace ocasionalmente. La forma de proceder más habitual en la reutilización de requisitos es utilizándolos como base y actualizándolos según cambian las necesidades particulares de un proyecto a otro. Los resultados se reflejan en la figura 73.



Figura 73: Reutilización de requisitos

Aspectos como la fiabilidad, tiempo de respuesta de la aplicación, mantenibilidad y disponibilidad son contemplados en los requisitos en un 75% de los casos.

3.3.3. **Datos sobre procesos de verificación y validación de los requisitos:**

En cuanto a las respuestas ofrecidas a las preguntas realizadas sobre procedimientos de verificación y validación de los requisitos se generan estas deducciones:



Un 16,7% realizan una fase de revisión de requisitos como parte del ciclo de vida de todos sus proyectos, un 66,7% lo realiza solamente en algunos de ellos y el restante no incluyen fase de revisión tal y como se muestra en la figura 74.

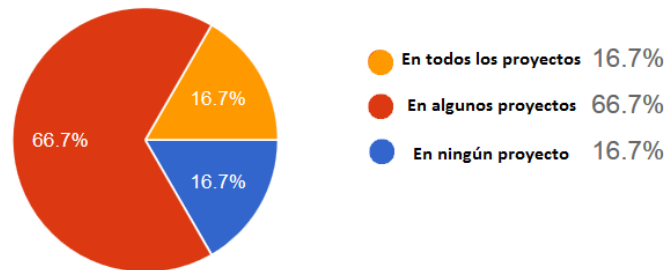


Figura 74: Aplicación de fase de revisión

La distribución de la participación de los roles más utilizados en el proceso de revisión se ajusta a lo descrito en la figura 75.

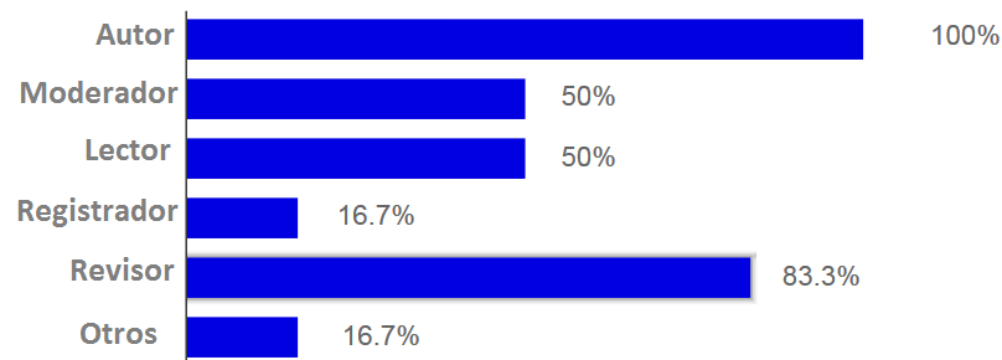


Figura 75: Distribución de los participantes en los procesos de revisión

Como puede observarse el autor participa en todos los procesos de revisión, acompañado en el 80% de las ocasiones por otra persona que ejerce como revisor. Solo en la mitad de los casos el proceso se sigue guiado por un moderador y un lector que expone los requisitos a los revisores y los autores.

Una vez detectado un error en una revisión y corregido, el 60% de los encuestados analizan si esa corrección tiene un impacto imprevisto en otros requisitos.

Se ha preguntado si se realiza un estudio sobre la viabilidad de llevar a cabo un requisito con el tiempo y costes pactados antes de delimitarlo en la línea base siendo así en un 85% de los casos.

Los procesos de validación y verificación pueden ser realizados por expertos que verificarán que el requisito se ciñe a las normas de realización y a las necesidades del cliente y validarán que la solución que propone es válida, también pueden ser automatizados mediante herramientas de verificación. En un 40% de los casos la



validación y verificación es realizada por expertos, mientras que en otro 40% es llevada a cabo por expertos ayudados de herramientas adicionales. En ningún caso es un proceso exclusivamente automatizado.

Estos resultados son mostrados de forma gráfica en la figura 76.

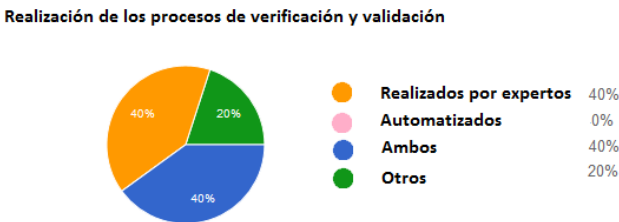


Figura 76: Modo realización proceso de verificación y validación

Se ha consultado a los encuestados por la proporción de tiempo del total del desarrollo que se dedican a las fases de verificación y de validación de los requisitos siendo utilizados de media un 5% del tiempo en la fase de verificación de requisitos y un 8% en la fase de validación de requisitos.

3.3.4. **Datos sobre procesos de gestión de los requisitos:**

En este apartado se han realizado preguntas sobre la gestión y mantenimiento de los requisitos una vez que se han redactado y validado.

Una de las principales responsabilidad de la gestión de requisitos es el manejo de las modificaciones que pueden sufrir por el cambio o evolución de las necesidades a lo largo del ciclo de vida del proyecto lo cual desemboca en la modificación de un requisito antes de que éste haya sido desarrollado por completo.

El proceso más común para manejar este tipo de procedimientos según los encuestados es el siguiente:

- 1º - Petición formal de la modificación por el cliente.
- 2º - Análisis de la modificación.
- 3º - Aprobación o desestimación de la modificación junto al cliente.
- 4º - Modificación final de la especificación del requisito al que hace referencia.
- 5º - Modificación de los componentes relacionados con el requisito.

En el 84% de los casos cualquier tipo de modificación queda documentada y archivada.

Se ha preguntado también si las empresas tenían alguna vía de comunicación para la transmisión de la modificación a todas las personas implicadas en los requisitos afectados contestando positivamente el 68% de los encuestados.



También nos hemos interesado por los procesos de auditoria destinados a comprobar que los requisitos se ciñen a las necesidades del cliente y a las normativas a las que se debe adaptar el proyecto. En este aspecto un 80% de las empresas realizan estos procesos de auditoria, no obstante no lo hacen en cada uno de sus proyectos si no solamente en los más críticos.

3.3.5. **Datos el uso de herramientas para la gestión de los requisitos:**

Se ha consultado a los encuestados sobre el uso de herramientas para la elaboración y gestión de los requisitos, herramientas para la trazabilidad de cambios de sus requisitos, herramientas de análisis semántico de los requisitos y herramientas de validación y verificación de requisitos.

Los datos obtenidos son mostrados en la figura 77 ilustrado el porcentaje de uso de cada tipo de herramienta por las empresas encuestadas.

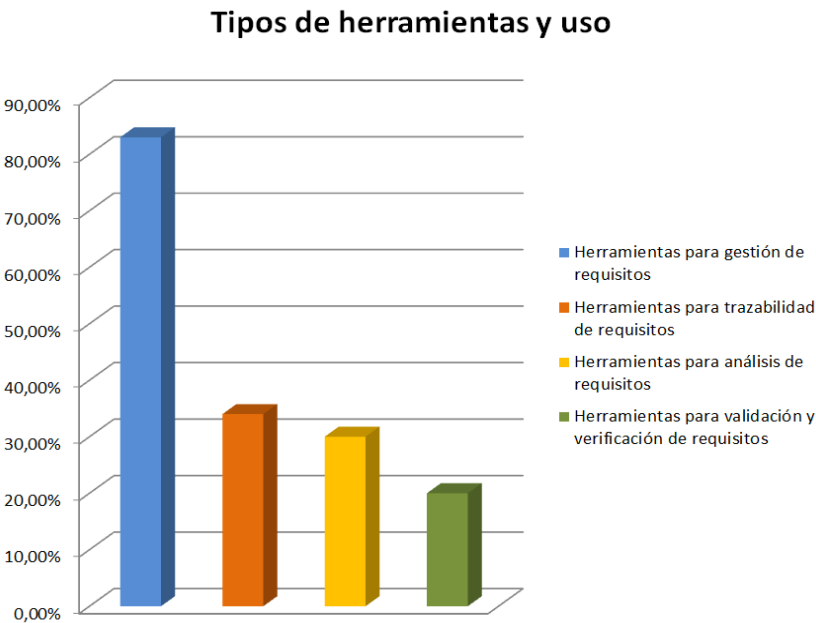


Figura 77: Tipos de herramientas y utilización

Las herramientas más utilizadas por los participantes del cuestionario se distribuyen en el gráfico de la figura 78 siendo IBM Rational DOORS la más habitual.

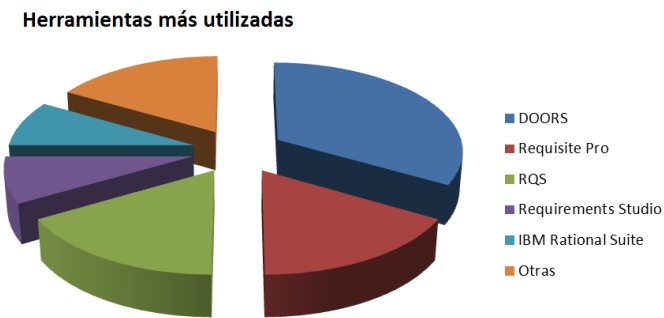


Figura 78: Herramientas más utilizadas

### 3.4. Conclusiones sobre el cuestionario:

Tras analizar los datos obtenidos en el cuestionario se deduce que prácticamente todas las empresas utilizan métodos para mejorar la calidad de sus requisitos en menor o mayor medida dependiendo de la criticidad del proyecto a desarrollar, es decir, si se trata de un desarrollo de un proyecto complejo o con riesgos importantes la aplicación de éstas medidas se intensifica. Sin embargo las empresas dedicadas a productos menos críticos tienden a descuidar los procesos de mejora y control de la calidad de sus requisitos.

La aplicación de procesos automatizados para la mejora de la calidad en los requisitos, tales como herramientas de análisis semántico, aún no está muy extendida aplicándose solo en empresas muy especializadas o innovadoras. La práctica habitual es confiar en el criterio de los expertos que generan y gestionan los requisitos apoyados en metodologías y normativas, utilizando por lo general, herramientas que ayudan a documentar y almacenar la información sobre los requisitos.

En definitiva, el uso de prácticas para la gestión de la calidad de los requisitos depende del tipo de producto a desarrollar y de la capacidad de innovación de las empresas no siendo aún un proceso interiorizado dentro de todos los ciclos de desarrollo.

## 4. Conclusiones y trabajo a futuro:

En este capítulo se mostrarán las conclusiones generales adquiridas tras la realización del mismo así como líneas futuras de trabajo para su ampliación y actualización.

### 4.1. Conclusiones generales:

Este proyecto fin de carrera se realizó con la intención de mostrar que vías existen para mejorar aumentar y gestionar la calidad en la realización de los requisitos y sobre todo crear conciencia acerca de lo importante y beneficioso que es invertir tiempo y recursos en este aspecto frente a lo perjudicial que resulta descuidar los requisitos.

Los datos recabados por diferentes estudios [3] confirman que una inversión de tiempo y recursos en aumentar la calidad de la documentación y gestión de los requisitos, acorde a la complejidad del producto a desarrollar, reduce significativamente los errores y contratiempos surgidos en las fases tardías ahorrando así los costes necesarios para subsanar dichos problemas en el tiempo establecido.

Dedicar esfuerzo en aumentar la calidad de los requisitos es aumentar la probabilidad de finalizar con éxito un proyecto (en tiempo y coste previstos) y ganar la confianza tanto por parte del cliente y del mercado como de los propios desarrolladores al trabajar en proyectos bien delimitados y gestionados.

Los compendios de buenas prácticas o metodologías, tales como ITIL o CMMI (hasta el grado 3), son una forma efectiva y relativamente asequible de introducir dentro de todos los desarrollos llevados a cabo por una empresa u organización, asegurando así unos niveles de calidad aceptables y dando paso a incorporar nuevas mejoras una vez se ha interiorizado su utilización dentro de los procesos de desarrollo.

Existen multitud de normativas cuyo objetivo es la mejora de la calidad en el desarrollo de productos. Muchas de ellas son específicas para la realización de un determinado tipo de producto (vehículos, navíos, sistemas de defensa...etc.) no obstante engloban prácticas comunes en cuanto a la realización de los requisitos se refiere. Las normas incluidas en este trabajo ISO/IEC/IEEE 12207:2008 y ISO/IEC/IEEE 15288:2008 incluyen una puesta en común de muchas de estas recomendaciones para conseguir requisitos con un alto nivel de calidad.

La inclusión de la fase de revisión como parte del ciclo de vida de un producto es una práctica efectiva y asequible en los costes derivados de su utilización para detectar y solventar errores e incongruencias dentro de los requisitos. Cada hora dedicada a corregir el contenido de los requisitos se transforma en 10 horas de ahorro en subsanar errores que no hubieran sido detectados sin la aplicación de esta fase.

El desempeño de las revisiones de requisitos es un proceso flexible que puede necesitar de un número de miembros variable según la cantidad y complejidad de los requisitos que se vayan a revisar pudiendo así adaptar el número de recursos según sea necesario.

Para la aplicación de todos estos métodos, prácticas y normas los desarrolladores pueden ayudarse con una gama cada vez mayor de herramientas destinadas a la realización y gestión de los requisitos. El desarrollador tiene a su disposición desde las herramientas simples que ayudan en gran medida a la generación, almacenaje y visualización de los



requisitos; herramientas en la nube que basan su efectividad en la cooperación de todos los miembros implicados en el proyecto y facilitando así el desarrollo conjunto a tiempo real. En estos últimos años la innovación en este tipo de herramientas está adquiriendo cada vez más importancia. Basando su potencia en la gestión del conocimiento y el uso del lenguaje disponemos de herramientas capaces de generar requisitos utilizando el conocimiento previamente adquirido previamente mediante el uso de repositorios de conocimiento y ontológicas. Este modo de generar requisitos con un lenguaje acotado y determinado permite utilizar herramientas, que mediante el análisis semántico del contenido de los requisitos, nos ayudan a corregir cualquier tipo de error en la realización de su contenido consiguiendo así alto niveles de calidad y facilitando y simplificando su comprensión.

Aunque aún parece que es práctica exclusiva de las empresas más especializadas o dedicadas al desarrollo de productos complejos, poco a poco las empresas ponen medidas para mejorar la calidad de sus requisitos porque está demostrado que es mejor (y más barato) prevenir que curar.

#### 4.2. Trabajo a futuro:

Teniendo en cuenta que se trata de un proyecto cuya finalidad es recopilar información sobre técnicas y métodos para la mejora de la calidad de los requisitos las líneas para su evolución en un futuro son:

- Ampliar la información sobre nuevos compendios de buenas prácticas.
- Ampliar la información sobre con normativas más específicas enfocadas en la realización de los requisitos.
- Incluir y estudiar nuevas herramientas para la generación y gestión de requisitos.
- Estudiar la línea de herramientas basadas en la gestión del lenguaje y el conocimiento como forma innovador de generación y gestión de los requisitos.
- Realizar una nueva encuesta sobre buenos y malos hábitos en la realización de los requisitos pasados unos años.

## 5. Glosario de términos:

**Requisito:** Un requisito es una especificación de un elemento o componente que debe ser realizado para la obtención de un producto o servicio final.

**Informes CHAOS:** Es un informe en el que se muestran los datos sobre éxitos y fracasos de diversas empresas del sector de TI. Este informe se suele llevar a cabo cada 2 años.

**Método SCAMPI:** El SCAMPI (Standard CMMI Appraisal Method for Process Improvement) es un método desarrollado por Instituto de Ingeniería de Software (SEI) para evaluar el estado de los procesos de software de una organización basado en los modelos CMMI.

**SEI: Software Engineering Institute** (SEI) es un instituto federal estadounidense de investigación y desarrollo, fundado por el Congreso de Estados Unidos en 1984 para desarrollar modelos de evaluación y mejora en el desarrollo de software, que dieran respuesta a los problemas que generaba al ejército estadounidense la programación e integración de los sub-sistemas de software en la construcción de complejos sistemas militares.

**TI:** Marco que agrupa lo relacionado con las Tecnologías de la Información. Las TI abarcan el dominio completo de la información, que incluye al hardware, al software, a los periféricos y a las redes. Un elemento cae dentro de la categoría de las TI cuando se usa con el propósito de almacenar, proteger, recuperar y procesar datos electrónicamente.

**ERS (Especificación de Requisitos Software):** Documento o conjunto de documentos en los que se describe el conjunto de los requisitos necesarios para el desarrollo de un producto o servicio.

**Reworking:** Fase del proceso de desarrollo en la cual se corrigen o modifican componentes implementados de forma errónea o incorrecta.

**Stakeholder:** Término inglés que designa a todas aquellas personas u organizaciones que afectan o son afectadas por un proyecto, ya sea de forma positiva o negativa.

**Educción:** La educación de requisitos consiste en hallar e identificar los requisitos que deben satisfacer un determinado producto o servicio. Se trata de una actividad propia de la ingeniería del software, anterior al análisis de requisitos.

**SaaS( Software-as-a-Service):** es un modelo de distribución de software donde el soporte lógico y los datos que maneja se alojan en servidores de una compañía de tecnologías de la información y comunicación (TIC), a los que se accede vía Internet desde un cliente. La empresa proveedora TIC se ocupa del servicio de mantenimiento, de la operación diaria y del soporte del software usado por el cliente. Regularmente el software puede ser consultado en cualquier computador, se encuentre presente en la empresa o no. Se deduce que la información, el procesamiento, los insumos, y los resultados de la lógica de negocio del software, están hospedados en la compañía de TIC.

**JIRA:** es una aplicación basada en web para el seguimiento de errores e incidentes y para la gestión operativa de proyectos. JIRA también se utiliza en áreas no técnicas para la administración de tareas.

**Rally:** Rally Software es un proveedor de herramientas de gestión de proyectos basadas en la aplicación de metodologías ágiles.

**TFS:** es la plataforma de colaboración en el núcleo de la solución de administración del ciclo de vida de las aplicaciones (ALM) de Microsoft. TFS admite prácticas ágiles de desarrollo, varios IDE y plataformas de manera local o en la nube y le proporciona las herramientas que necesita para administrar de manera eficaz los proyectos de desarrollo de software a lo largo del ciclo de TI.

**HPQC:** HP Quality Center es una herramienta de gestión de pruebas cuyo objetivo es el control de la calidad de software, incluyendo la gestión de requisitos, gestión de pruebas y los procesos de negocio para entornos y aplicaciones IT.

**Microblogging:** Es un servicio que permite a sus usuarios enviar y publicar mensajes breves,<sup>1</sup> generalmente solo de texto, caracterizándose principalmente por su sencillez. El servicio de microblogging por excelencia es la plataforma Twitter.

**Ontología:** El término ontología en informática hace referencia a la formulación de un exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios dados; con la finalidad de facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades.

**Tesaurus:** Tesaurus es un sistema de organización del conocimiento, en forma de lista, que contiene los «términos» controlados empleados para representar los conceptos, temas o contenidos de los documentos, con miras a efectuar una normalización terminológica que permita mejorar el canal de acceso y comunicación entre los usuarios y las unidades de información.

**ISO 9000:** Es un conjunto de normas sobre calidad y gestión de calidad, establecidas por la Organización Internacional de Normalización (ISO). Se pueden aplicar en cualquier tipo de organización o actividad orientada a la producción de bienes o servicios. Las normas recogen tanto el contenido mínimo como las guías y herramientas específicas de implantación como los métodos de auditoría. El ISO 9000 especifica la manera en que una organización opera sus estándares de calidad, tiempos de entrega y niveles de servicio.

**ISO 20000:** La serie ISO/IEC 20000 – Gestión de servicios normalizada y publicada por las organizaciones ISO e IEC (International Electrotechnical Commission) el 14 de diciembre de 2005, es el estándar reconocido internacionalmente en gestión de servicios de TI (Tecnologías de la Información).

## 6. Bibliografías y referencias:

- 1- Guía Práctica de Gestión de Requisitos, INTECO, 2008.
- 2- El proceso de Ingeniería de Requisitos en el ciclo global del software, Rosa María Torres de Paz, Universidad de Sevilla, 2010.
- 3- Standish Group , Página Oficial , disponible en: <http://standishgroup.com> [Consulta: 20/10/2014]
- 4- Software Engineering Institute, Official site of CMMI, <http://www.sei.cmu.edu/cmmi/> [Consulta: 05/11/2014]
- 5- [http://es.wikipedia.org/wiki/Information\\_Technology\\_Infrastructure\\_Library](http://es.wikipedia.org/wiki/Information_Technology_Infrastructure_Library) [Consulta: 07/11/2014]
- 6- Karl E. Wiegers, More About Software Requirements, Microsoft Press, 2006, 2005936071
- 7- Spanish Technical Report CMMI V 1 3
- 8- Wiegers K., Beatty J. - Software Requirements 3, 3rd Edition – 2013.
- 9- Klawen, Área de procesos CMMI –Gestión de Requisitos, Octubre 2014.
- 10- Sitio oficial de ITIL, <https://www.axelos.com/best-practice-solutions/itil/itil-publications>. [Consulta: 02/12/2014]
- 11- Estado de la práctica de la Ingeniería de Requisitos en proyectos de Software O.S. – Yolanda Escribano Luis, Universitat Politècnica de Catalunya.
- 12- Managing Requirements & Improving Quality, Shambhavi Roy.
- 13- ISO/IEC/IEEE 29148 – 2011, Systems and software engineering – Life cycle processes – Requirements engineering.
- 14- ISO/IEC/IEEE 12207:2008 - Systems and software engineering – Software life cycle processes.
- 15- ISO/IEC/IEEE 15288:2008 – Systems and software engineering – Systems life cycle processes.
- 16- ISO/IEC/IEEE 15289:2011 – Systems and software engineering – Content of life-cycle information products (documentation).
- 17- ISO/IEC TR 19759 – Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOOK)
- 18- IEEE Std 830 – IEEE Recommended Practice for Software Requirements Specifications.
- 19- IEEE Std 1233 – IEEE Guide for Developing System Requirements Specifications
- 20- ISO/IEC TR 24748-1 – Systems and software engineering – Life cycle management – Part 1: Guide for life cycle management.
- 21- ISO/IEC/IEEE 24765 – Systems and software engineering – Vocabulary.
- 22- <http://docs.innoslate.com/latest/users-guide/> [Consulta: 08/02/2015]
- 23- [http://en.wikipedia.org/wiki/Jama\\_Software](http://en.wikipedia.org/wiki/Jama_Software) [Consulta: 14/03/2015]
- 24- <http://www.jamasoftware.com> [Consulta: 14/03/2015]
- 25- <http://www.reusecompany.com> [Consulta: 18/04/2015]
- 26- <http://www.ciset.es/> [Consulta: 22/04/2015]
- 27- <http://www.incose.org/Home> [Consulta: 26/04/2015]
- 28- [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=51041](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51041) [Consulta: 12/03/2015]
- 29- <https://www.google.com/forms/about/>
- 30- <http://www.eduardoriol.com/cmmi-para-principiantes-y-iii/> [Consulta: 06/11/2014]